System User’s Guide
for the
EOSDIS Test System (ETS)
MPS/Aura Simulator

Release 3.0
January 2002

Prepared Under Contract NAS9-98100
by Computer Sciences Corporation

System User's Guide
for the
EOSDIS Test System (ETS)
MPS/Aura Simulator

Release 3.0
January 2002
Prepared Under Contract NAS9-98100
by Computer Sciences Corporation

Prepared by:

____
J. K. Carlson, CSC Date

Reviewed by:

____
E. Quintin Date
ETS Technica1 Lead, CSC

____
E. Noone Date
Simulations Technical Area Manager, CSC

Approved by:

____
W. Fuller Date
ETS Manager, Code 581

About This Document

This document is the System User's Guide for the Scalable, Integrated, Multimission, Simulation Suite (SIMSS) Aura Simulator developed for the National Aeronautics and Space Administration (NASA) Goddard Space Flight Center (GSFC). This simulator is commonly referred to as the MPS/Aura simulator. The simulator is one of the deliverables of the EOSDIS Test System (ETS) Development Task under the CSOC contract. The System User's Guide provides an overview of operational concepts and procedures for this simulator. This document is approved by GSFC Code 581, and will be maintained by Computer Sciences Corporation (CSC) under the CSOC contract.

Who Should Use This Document

This document is for anyone who wishes to use the MPS/Aura Simulator. To use this simulator effectively you should already be familiar with the basics of the Windows NT Operating System as well as be familiar with the EOS Aura Spacecraft.

Document Overview

The MPS/Aura simulator is built with SIMSS components. It contains several generic modules: InputIP, OutputIP, LogModule, SerialInput, SerialOutput, Scenario, Txfile and VC Processor. It also has two modules, EOSGS and SCAura, that were customised for the EOS Aura mission. This document has been assembled with SIMSS generic and Aura specific information.

The first chapter provides a complete description of the SIMSS client and server project user interface. It describes the core operations of building, configuring, saving, restoring and running a project. All users should read this chapter to become familiar with SIMSS terminology and the common project user interface that will be referenced in later chapters.

Next are chapters that describe each of the modules that have been delivered as components of MPS/Aura. For each module, there is information on how to link it to other SIMSS modules, how to configure it, and how to control its run-time operations.

Chapter	Module Name	Description
EOSGS	EOSGS	EOS Ground Station Module
SCAura	SCAura	Spacecraft Module for Aura
IP	InputIP, OutputIP	Input and Output Internet Protocol Modules
Log	LogModule	Log Module
Scenario	Scenario	Scenario Module
SI	SerialInput	Serial Input Module
SO	SerialOutput	Serial Output Module
TxFile	TxFile	Transmit File Module
VC Processor	VC Processor	Virtual Channel Processor Module

How to Run a SIMSS-Based Simulator

The Scalable, Integrated, Multimission Simulation Suite (SIMSS) is a distributed, component-based, plug-and-play, client-server system useful for performing real-time simulations and communications testing in support of NASA projects.

SIMSS-1.0 Terminology

SIMSS runs on one or more Windows NT workstations. It is designed to be user-configurable or to use predefined configurations for routine operations.

Client: The workstation on which the user interface runs. It also refers to the applications making up the user interface.

Server: The workstation on which the actual data processing is performed. It also refers to the applications making up the data processing component.

Module: A self-contained SIMSS component that receives, processes, or transmits data, or any combination of the three.

Channel: An interface or port through which a module receives or transmits data. A module may have zero to many channels.

Link: A directional connection between module channels. A link connects an output channel of one module with the input channel of another module.

Project: A collection of modules and links intended to perform a specific function such as spacecraft simulation, data quality monitoring, or data conversion.

Event Message: A time-tagged text message generated by any module to inform the operator. Event messages may report warning or error conditions as well as successful activities.

Directive: A text-based command line entry that operators submit to individual modules. The directive entry line window is below the configuration window in the main SIMSS display.

Container: A repository internal to each module that contains all of the vital data for that module. The container is used to exchange data between the client and server. Built-in functions of the container support save and restore operations.

Scenario: A file that contains directives. A scenario module may execute this file in order to send a timed repeatable sequence of directives to a linked module.

SIMSS-2.0 Installing A New Release

SIMSS-2.1 Installing the Client

To install the SIMSS client, first run jdk1_2_2-win.exe located in the root of the delivery CD. After JDK1.2.2 is installed, the next step is to run setup.exe in the \client folder on the CD and follow the prompts. Upon completion of the client installation, a SIMSS Client icon will be installed on the desktop. The SIMSS Client icon will point to the executable directory for the client software. Below this directory are three additional directories: images, properties, and save.

Directory	Description
images	Contains bitmapped image files.
properties	Contains client configuration files.
save	This directory is where client information is stored when projects are saved. The user may want to delete old files from this directory. (Files with the same names should also be deleted from the Server save directory)

SIMSS-2.2 Installing the Server

To install the SIMSS server, run *setup.exe* in the *\server* folder on the CD and follow the prompts. Upon completion of the server installation, a SIMSS Server icon will be installed on the desktop. The SIMSS Server icon will point to the executable directory for the server software. Below this directory are four additional directories: elog, properties, save and scenario.

Directory	Description
elog	Contains event message log files. The user may want to delete old event message log files from this directory to reclaim disk space.
properties	Contains the server property.txt configuration file.
save	This directory is where server information is stored when projects are saved. The user may want to delete old files from this directory. (Files with the same names should also be deleted from the Client save directory)
scenario	Contains delivered scenario files. The user may store additional scenario files here.

SIMSS-3.0 Configuration Files

SIMSS-3.1 Client/Properties/Servers.txt

Before running the SIMSS Client, please check that the correct IP address for running the SIMSS Server is in the *properties/servers.txt* file. This file must be present and must contain the *localhost* entry. If the SIMSS Client is to be operated remotely from the SIMSS Server then additional entries, giving the IP addresses of each SIMSS Server desired, must be added to the file.

The format for entries in this file is:

Servername, IP address

Example:
localhost,127.0.0.1

SIMSS-3.2 Server/Properties/Property.txt

There are several parameters that may be configured in the server. Most of them are for debugging purposes. These parameters are stored in the *properties*/*property.txt* file. This file is read in each time that the server is started. The values used from the property.txt file are written into the server text window. The property.txt file contains descriptions of the parameters and their normal settings. Use caution in making modifications to this file. Some parameters that may be of interest to users are listed below.

Property	Description
EventMsgLogSize	This is a maximum event message log size. When this many event messages have been logged, the oldest messages will be overwritten. When set to zero, there is no limit on the number of event messages that are saved to the log.
DelOldLog	This flag is for deleting the old event log file on startup. (0=no, 1=yes). This is only effective if the event log file name is the same for each execution. Use with DefaultEventLogSave.
DefaultEventLogSave	This flag specifies use of the same event log file name for each execution (0=use different names, 1=use same name).
ConcurrentScenario	This flag applies to scenario modules in the project. The scenario mode may be set to 0 for serial operation or 1 for concurrent operation. It is normally set to 1. This parameter applies only to scenario files that are not under direct operator GUI control, such as scenario files started by other scenario files. If too many scenarios run concurrently, there could be system resource issues.

SIMSS-4.0 Running a SIMSS Application

The SIMSS modular architecture can support a variety of applications by connecting generic and mission-specific modules in different combinations. The general steps for starting up any SIMSS project are summarized in the next section. Detailed examples of each of these steps are provided in SIMSS-5.0, which describes the graphical user interface (GUI).

Every SIMSS application must run within a project. Each project consists of interconnected executable software modules. Every software module may have configuration and run-time displays. Every module that accepts operator directives can execute the directives described later in this chapter. Every module that executes directives may also be configured to accept directives from a scenario file. Every module that generates event messages for the operator sends them to the event log region of its project window.

Other chapters in this system user's guide are devoted to the various generic and mission-specific SIMSS modules. Please refer to these chapters for detailed information on module-specific configuration and run-time operations.

Paragraph SIMSS-4.1 provides an overview of the steps necessary to create or restore a project for operations. The subsections under paragraph SIMSS-5.0 provide more details about the steps where options are available.

SIMSS-4.1 Application Startup Sequence

Startup Steps	**Description**
1) Start the Server	Double-click the Server icon on the PC desktop. When the Server is started, a text window is created.
2) Start the Client	Double-click the SIMSS Client icon on the desktop. When the Client is started, a Client text window is created and then the graphical user interface is started with the SIMSS Client main window.
3) Add (or Restore) a Project	Click on the SIMSS Client main window's Project menu and select "Add Project". Or, a "Restore From" selection can be used to load a previously saved project. A project window is created within the main window and the project event message window is created. If a previously saved project is restored, skip to step 7.
4) Connect Project to Server	In the project window, click the System menu and select the "Connect" option. A list of available servers will be shown. Select a server for the connection. When you connect to the server, the message in the lower, left-hand corner of the project changes from "Status" to "Project Loaded."
5) Add Modules	In the project window, click the Module menu and select "Add Module". From the Module Selection window, click on the module to be added and the OK button. Click again in the project window for placement of the module's icon. Repeat this step until all desired modules have been added.

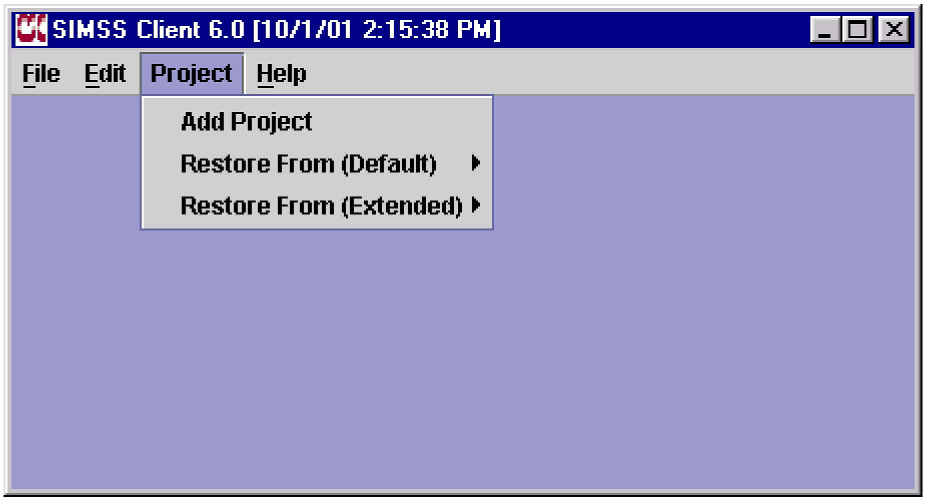
6) Create Links Note: Refer to the module specific chapters for input and output channel data descriptions in order to link the output channels to the appropriate input channels!	In the project window, click the Module menu and select “Create Links”. Click on the sending module border then move the cursor to the receiving module border and click again. If either module has multiple links, a popup window will appear for specification of the exact links. Repeat this step to create all desired links. To avoid having the link line go through intervening modules or overlay another link, click on the sending module, move the cursor to a convenient point on the Project background and click once, then move the cursor to the receiving module and click again. To cancel link creation mode, click on the Module menu and select “Design”.
7) Configure Modules	Click in the center of the module and select “Configure” from the popup menu. Perform module-specific configurations. Repeat for as many modules as needed.
8) Lock the Project	In the project window, click the Run menu and “Lock”.
9) Save the Project (optional)	Click the project window’s System menu and select a “Save Project” option. The Default option saves to the local PC. The Extended option allows navigation and saving to a networked PC. Specify a name for the project save file and apply it from the save or browse window.
10) Run the Project	In the project window, click the Run menu and select “Run”. All of the modules in the project will be started. Click in the center of any module to access its run-time options and displays.

SIMSS-5.0 SIMSS Application Displays

This section describes the graphical user interface common to all SIMSS applications. Every application has graphical client and project windows. Every module that accepts operator directives can execute the directives described later in this SIMSS chapter. Every module that generates event messages sends them to the project event message window.

SIMSS-5.1 SIMSS Client Main Window

When the SIMSS Client is started, the main window appears.



SIMSS-5.1.1 File Menu

The File menu contains an "Exit" option.

SIMSS-5.1.2 Edit Menu

The Edit menu contains a "Look and Feel" option. When this option is chosen, the operator may select from "System", "Metal", "Motif", or "Windows". The default setting is "Metal".

SIMSS-5.1.3 Project Menu

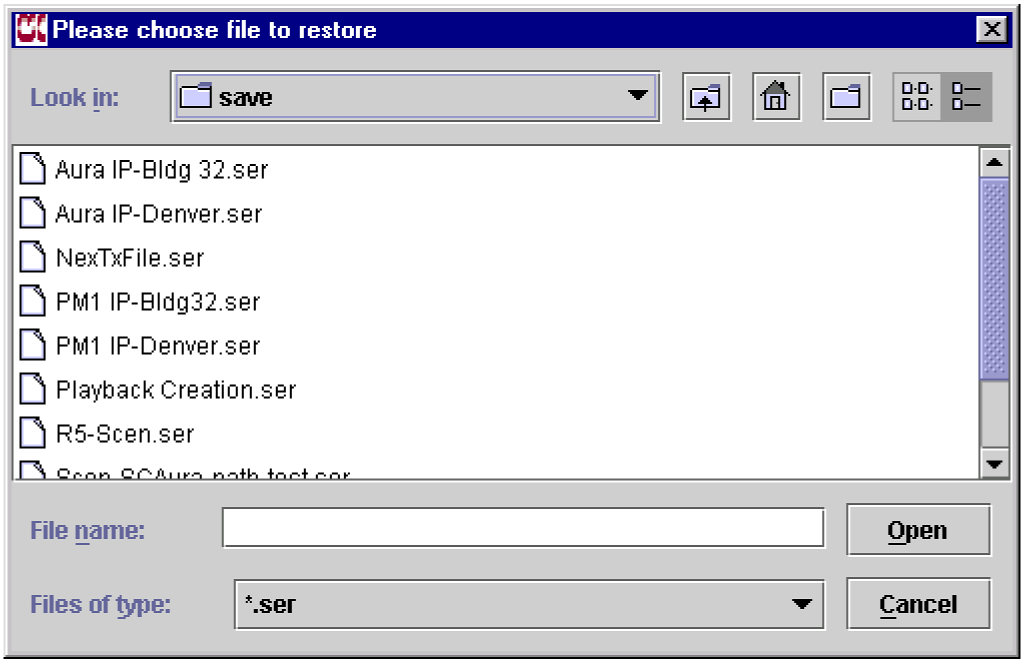
The Project menu contains three selections, "Add Project", "Restore From (Default)", "Restore From (Extended)".

SIMSS-5.1.3.1 Add Project

To create a new project, select the "Add Project" item from the Project menu. This will add a project window within the SIMSS client window and open the separate project event message window.

SIMSS-5.1.3.2 Restore From (Default)

To restore an existing project from the local PC, select the "Restore From (Default)" item from the Project menu. A file selection screen like the following will appear.

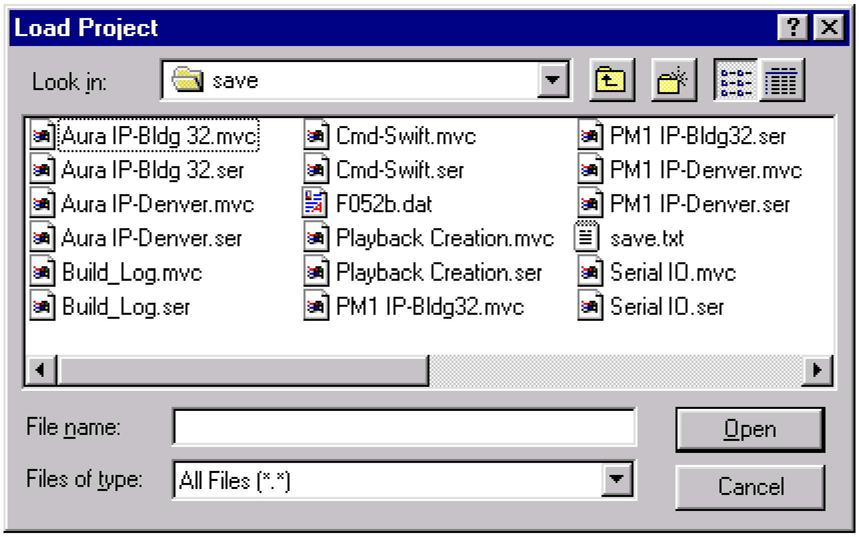


Aura IP-Bldg 32.ser
Aura IP-Denver.ser
NexTxFile.ser
PM1 IP-Bldg32.ser
PM1 IP-Denver.ser
Playback Creation.ser
R5-Scen.ser
Scen SCAura path test.ser

The system will search the save directory by default but another directory may be selected by clicking the Save Folder button. Select a file and click the **Open** button. Click on the **Cancel** button to dismiss the display without restoring a project.

SIMSS-5.1.3.3 Restore From (Extended)

To restore an existing project from another PC within the same network, select the "Restore From Extended" item from the Project menu. A file selection screen like the following will appear. This file selection window allows navigation to directories within the network neighborhood.



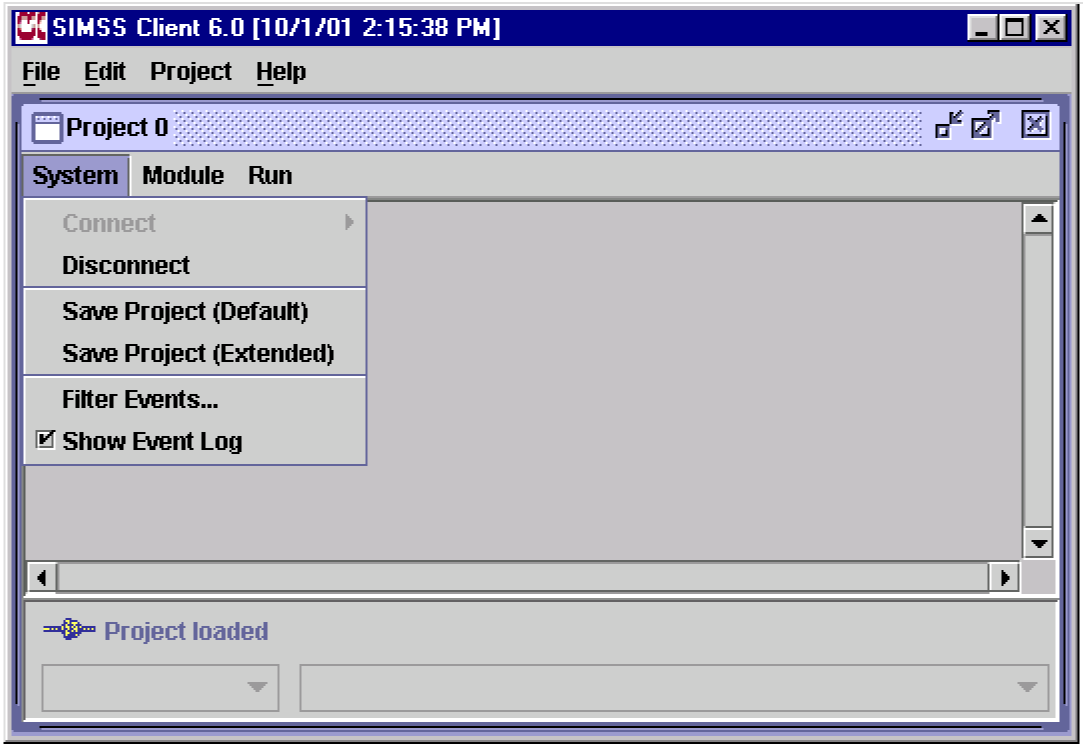
Aura IP-Bldg 32.mvc
Aura IP-Bldg 32.ser
Aura IP-Denver.mvc
Aura IP-Denver.ser
Build_Log.mvc
Build_Log.ser
Cmd-Swift.mvc
Cmd-Swift.ser
F052b.dat
Playback Creation.mvc
Playback Creation.ser
PM1 IP-Bldg32.mvc
PM1 IP-Bldg32.ser
PM1 IP-Denver.mvc
PM1 IP-Denver.ser
save.txt
Serial IO.mvc
Serial IO.ser

SIMSS-5.1.4 Help Menu

The Help menu contains an "About" option that provides client version information.

SIMSS-5.2 Project Window

When a project is added (or restored) its project window is displayed and its Project Event Message window (see SIMSS-5.3) is displayed.



Project loaded

The project window contains a System menu, a Module menu and a Run menu. At the bottom left of the project window is a status field. At the bottom right is the directive region of the screen.

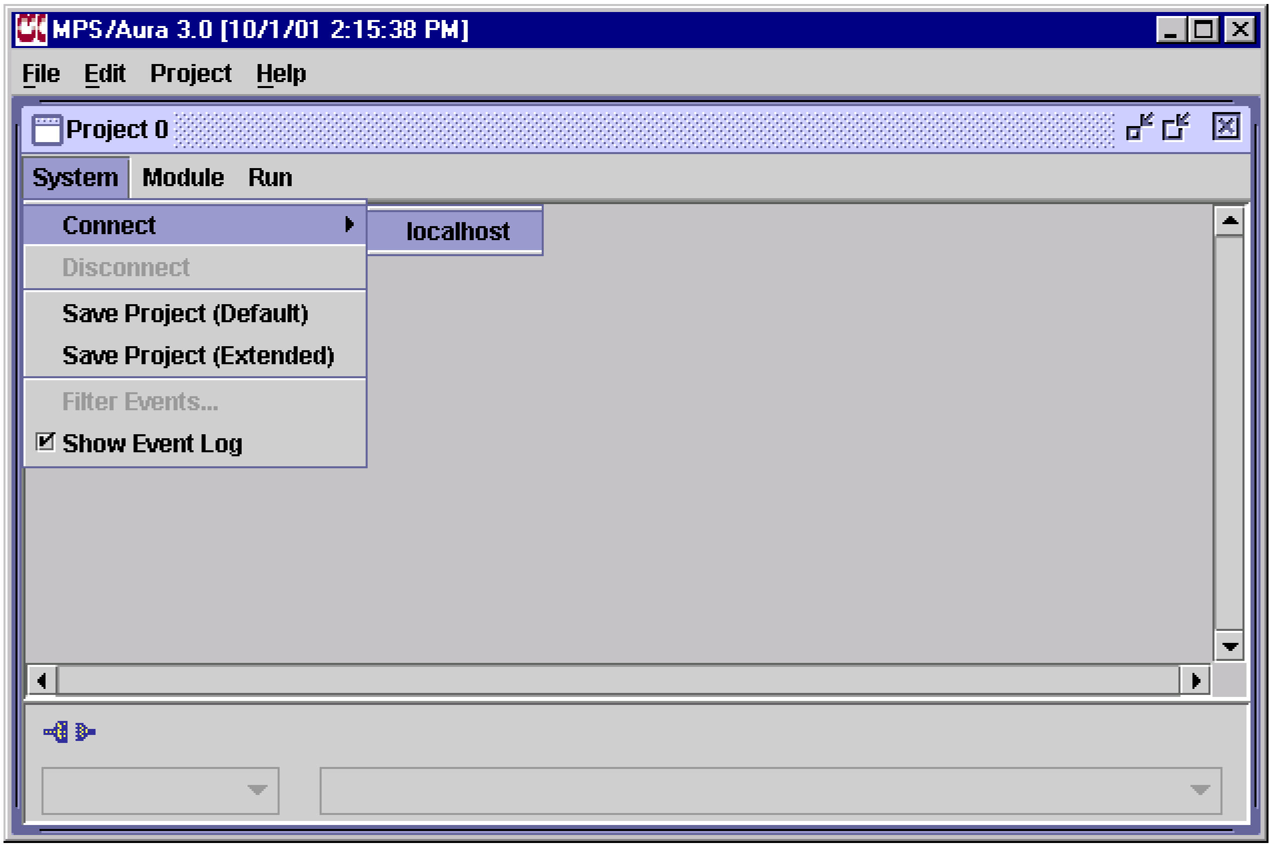
SIMSS-5.2.1 Project System Menu

The System menu on the Project window contains the following choices.

System Menu Item	Description
Connect	Connects the project to a server
Disconnect	Disconnects the project from a server
Save Project (Default)	Saves project design in file on local PC
Save Project (Extended)	Saves project design to file in PC network
Filter Events…	Change event message filtering
Show Event Log	When this option is enabled, the project event message log window is displayed.

SIMSS-5.2.1.1 Connect to Server

The client must be connected to a server in order to function. To make a connection to a server, select the "Connect" option from the System menu. A menu of available servers will be displayed. Click to select a server for the connection. In the following picture, only one server was available.



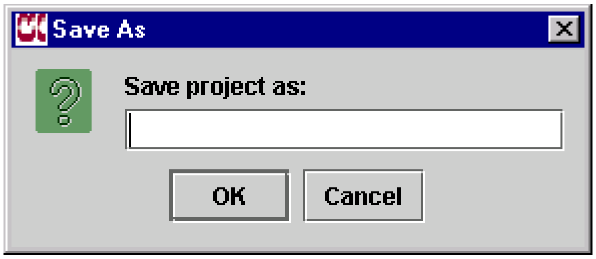
After the client has been connected to a server, the "Disconnect" option will be enabled on the System menu and the "Connect" option will be disabled.

SIMSS-5.2.1.2 Disconnect from Server

To disconnect the client from the server, select the "Disconnect" option from the System menu. After the client has disconnected from the server, the "Disconnect" option will be disabled on the System menu and the "Connect" option will be enabled.

SIMSS-5.2.1.3 Save Project (Default)

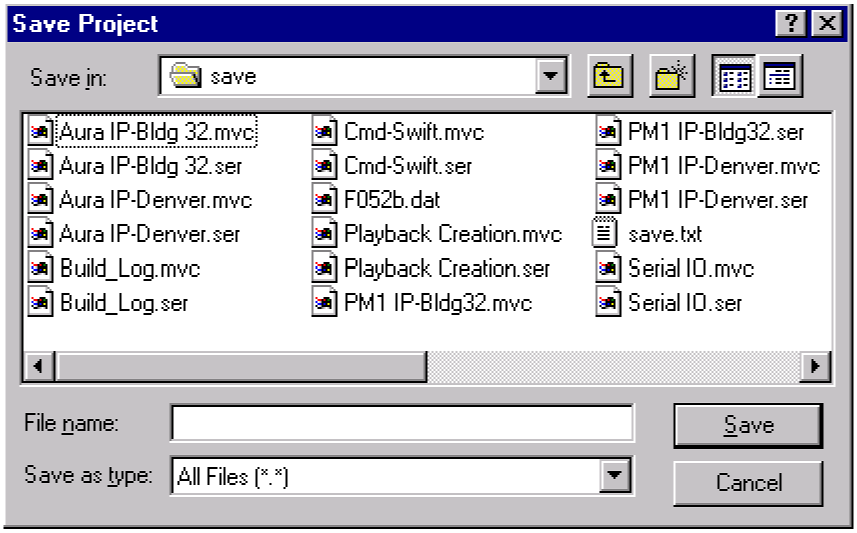
This option allows the user to store a created and configured project to a file on the local PC. The following screen will appear.



Enter the file name for storage of the current project and click the **OK** button to have project information written into it. The file name will have ".ser" appended to the end. Clicking **Cancel** will close this window without performing a save.

SIMSS-5.2.1.4 Save Project (Extended)

To save a project on another PC within the same network, select the "Save Project (Extended)" option from the Project System menu. A file selection screen like the following will appear. This file selection window allows navigation to directories within the network neighborhood.

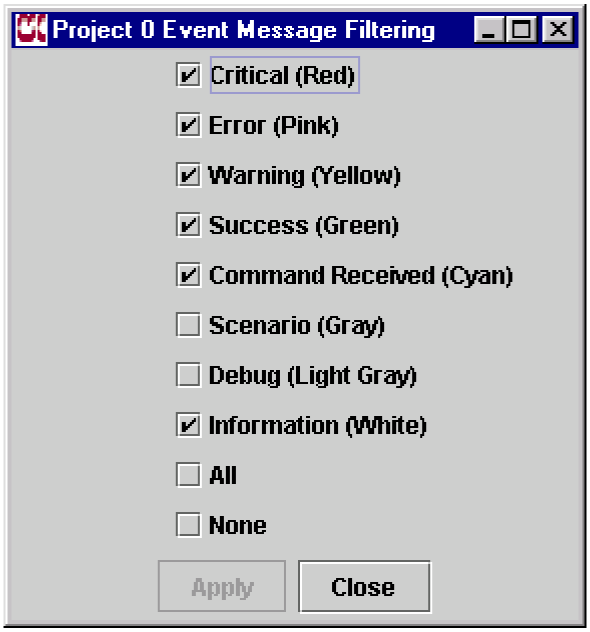


Save in: save

Save as type: All Files (*.*)

SIMSS-5.2.1.5 Event Message Filtering

When "Filter Events…" is selected, the Event Message Filtering screen appears. This image shows the default filtering scheme. Only event messages with the selected colors will be displayed in the event message window.



Project 0 Event Message Filtering

- ☑ Critical (Red)
- ☑ Error (Pink)
- ☑ Warning (Yellow)
- ☑ Success (Green)
- ☑ Command Received (Cyan)
- □ Scenario (Gray)
- □ Debug (Light Gray)
- ☑ Information (White)
- □ All
- □ None

<u>All event messages will continue to be generated and written into the event message log file.</u>

To disable the display of any of these event message types, clear its checkbox.

To enable the display of any of these event message types, check its checkbox.

Use the **Apply** button for changes to take effect.
Use the **Close** button to dismiss this screen.

<u>SIMSS-5.2.1.6 Show Event Log</u>

Each project has an event message log that contains color-coded, time-tagged event messages received from its modules. When the Show Event Log box is checked the event log is displayed as a separate scrolling window. When this box is cleared, the event window is closed. Closing the event window does not affect the writing of event messages to the Event Message Log file.

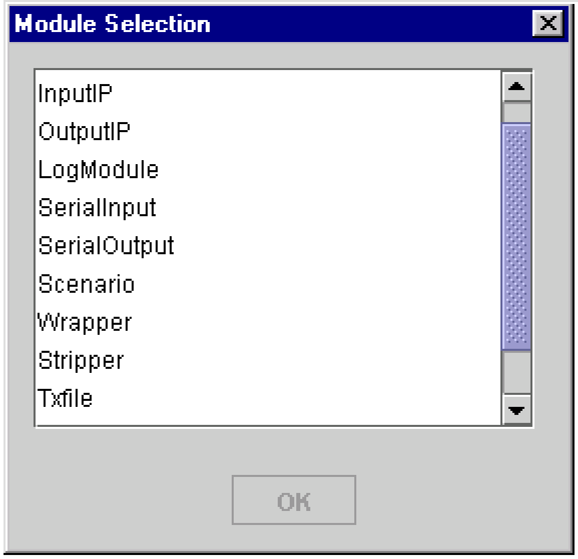
SIMSS-5.2.2 Project Module Menu

The Module menu on the Project window contains the following choices.

Module Menu Item	Description
Add Module	Add a module to the project's design diagram
Create Links	Create a link between modules
Design	Allows editing of the project design diagram

<u>SIMSS-5.2.2.1 Add Module</u>

To add a module to the project, select the "Add Module" option from the Module menu.



A list of the modules available from the server will be displayed. Click to select a module and then click the **OK** button.

Position the cursor within the project window where the top left corner of the module's symbol should be drawn and click. A rectangle representing the module will be drawn at that location. The rectangle will have a wide cyan border while in Design mode.

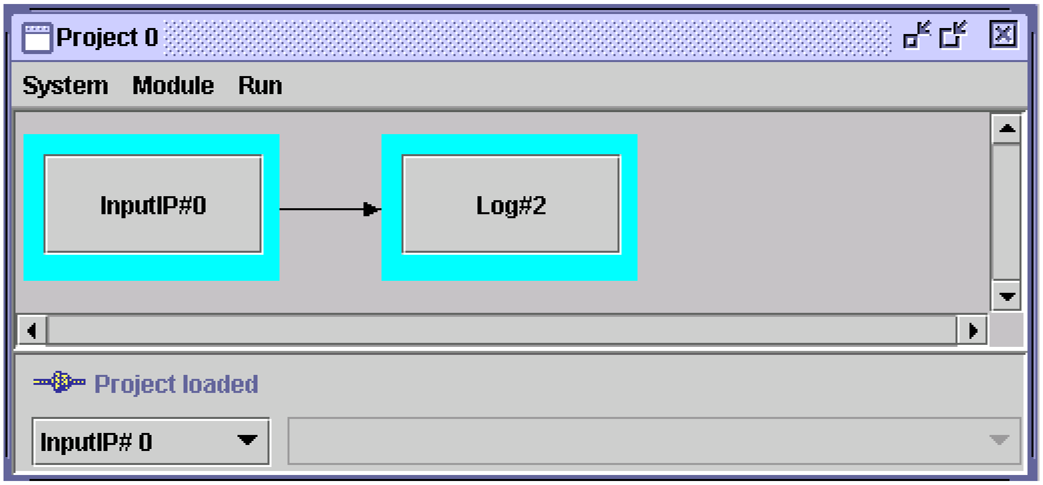
If a module is added that accepts operator directives, the directive region of the project window becomes accessible.

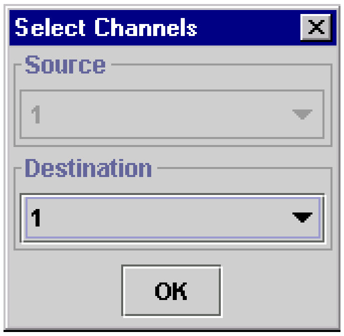
SIMSS-5.2.2.2 Create Links

Links are used to connect the input and output channels between modules for a given project. In the example shown below, the output channel of an Input IP module is linked to the input channel of a Log module.

To create a link,

- select the "Create Links" option from the Module menu,
- position the crosshair cursor on the highlighted edge of the source module and click,
- move the cursor to the destination module's edge and click again.





Source
1

Destination
1

When multiple links are defined for a module, the user will be prompted to choose a channel number for the source and destination links as appropriate. Refer to module-specific information for the number of input and output channels a module can have and how they should be configured.

The cursor remains in Create Links mode to allow for the creation of additional links. The cursor appears as a crosshair symbol until a different module menu option is selected.

If multiple links are created between two modules, additional clicks may be done on the project background to anchor the link lines apart for visibility.

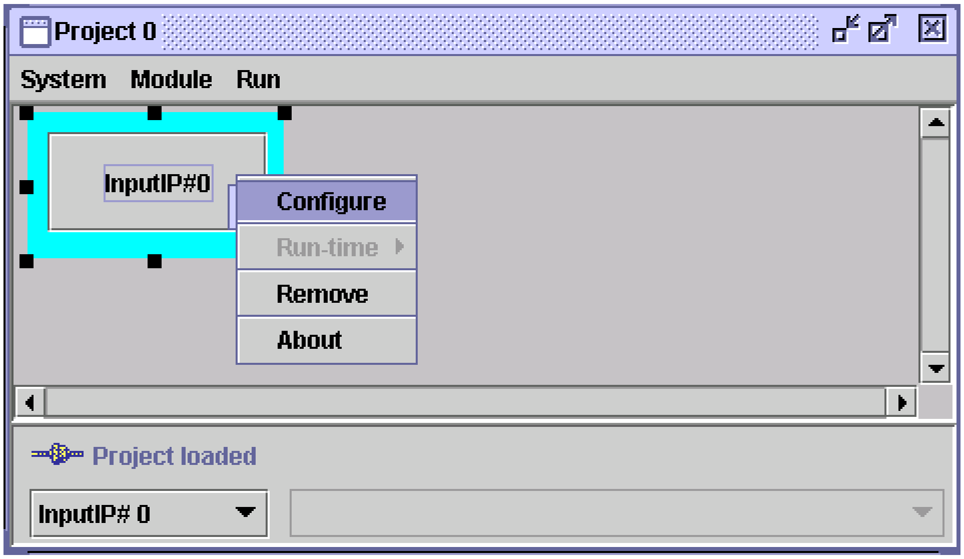
SIMSS-5.2.2.3 Design Mode

After the "Create Links" option has been used, the cursor remains as a crosshair symbol and the system remains in link creation mode. To cancel this mode, click the "Design" option from the Module menu. The cursor is changed back into a pointer symbol and may be used to select items of the design diagram for modification. While in this mode the operator may click on a link line to review the channel numbers of the modules it connects. A popup window provides the source and destination channel numbers. While a link line is selected, any of its background anchors may be moved by clicking and dragging.

SIMSS-5.2.2.4 Module Pop-Up Menu

Clicking in the center of a module activates a pop-up menu with the following choices.

Module Pop-Up Menu	Description
Configure	Provides access to module-specific configuration options.
Run-time	Provides access to module-specific run-time options.
Remove	Removes the module from the project.
About	Provides information about the module.



The actions initiated by the "Configure" and "Run-time" options are different for every module and are described in the other chapters of this system user's guide that are dedicated to specific modules.

SIMSS-5.2.3 Project Run Menu

The Run menu on the Project window contains the following choices.

Run Menu Item	Description
Lock/Unlock	Locks/Unlocks the project's design diagram
Run/Stop	Starts/Stops the project's execution

SIMSS-5.2.3.1 Locking a Project

Prior to running a project, the user must lock its design by clicking the "Lock" option from the Run menu. After the design is locked, the "Lock" option is replaced by the "Unlock" option and the "Run" option becomes available on the Run menu.

SIMSS-5.2.3.2 Unlocking a Project

Click the "Unlock" option from the Run menu to allow modification to the project's design diagram. The "Unlock" option is then replaced on the Run menu with a "Lock" option.

SIMSS-5.2.3.3 Running a Project

Click the "Run" option from the Run menu to start running all of the modules of the project. The borders of the modules will change to green to indicate run mode. The "Run" option is then replaced with a "Stop" option.

Click on specific modules to get their pop-up menus. See paragraph SIMSS-5.2.2.4 for an example. Click on the Run-time option for module-specific displays or actions.

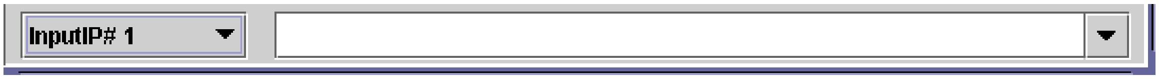
Please refer to the module-specific chapters in this system user's guide for information on each module's configuration and run-time displays.

SIMSS-5.2.3.4 Stopping a Project

Click the "Stop" option from the Run menu to stop execution of the modules of the project. The borders of the modules will change to red to indicate the stopped condition. The "Stop" option is then replaced with a "Run" option on the Run menu.

SIMSS-5.2.4 Project Directive Region

If any module in a project accepts operator directives, the directive region in the lower right side of the project window frame is activated. The directive region has a module indicator button and a data entry field. When the module indicator button is clicked, a list of the modules accepting directives is shown. Click to select the module to process a directive. The button is labeled with the currently selected module. In the example below, the Input IP module is selected. Click in the data entry field to the right of the module button to start entering a directive. Alternatively, click on the down arrow to the right of the field to view up to ten previously entered directives. A previous entry may be selected, optionally edited, and submitted. Use the keyboard's enter key to send the directive to the module. Directives are available to any module that accepts directives and has container variables.



SIMSS-5.2.4.1 Directives/Scenario Files

This section describes all of the directives currently defined for all SIMSS modules that accept operator directives. These directives may be placed in text files to be used as scenario files. Any module that processes directives can also receive directive lines read from a scenario file by the Scenario module. In addition to the directives described in this section, there are directives that are executed only by the Scenario module. These scenario directives control the timing and conditional execution of scenario directive lines. Please see the chapter on the Scenario module for more information on creating and executing scenario files. Refer to the container descriptions in each module's chapter for the names (mnemonics) of container items for each module.

SIMSS-5.2.4.2 Set Directive

The **set** directive may be used to change the value of any modifiable variable in the container for the indicated module. The format of the **set** directive is

```
set name value
```

where *name* is a container variable name and *value* is either a decimal, octal, or hexadecimal number. Octal numbers are identified by a leading zero. Hex numbers are identified by a leading 0x. Negative numbers may be entered as signed decimal integers in the usual manner, provided that the container chosen accepts negative numbers. Example: Telemetry points that are 2’s complement integers.

The names of variables are provided in the container description for each module. If the variable is successfully set, an event message of the form

“module: *name* set to *value*”

informs the operator of the change. If the variable name is not recognized or the variable cannot be modified, an event message of the type

“module: *name* invalid, read-only or not found“

will inform the operator. If unsuccessful, verify that the module indicator is correct, that the variable name spelling is correct and that the variable is not defined as read-only.

SIMSS-5.2.4.2.1 Set directive using expressions

More complicated assignments of values to container items are possible. Telemetry mnemonics (or any other type of container item) may be set to values derived from any combination of arithmetic expressions, arithmetic functions, Boolean functions, and other container items. These directives may be used to create sophisticated scenario files. The length of an individual set directive is limited to 100 characters. Intermediate variables have been created in part to allow writing of expressions that would otherwise be too long. In order to use any of the following expressions, specification of an equal sign in one of the following forms is required (=, +=, -=, *=, /=, ^=) after the mnemonic name and before the expression. The following sections provide syntax information and examples of set directives using expressions.

SIMSS-5.2.4.2.2 Set directive using arithmetic expressions

A container item may be set to an arithmetic expression using the symbols for addition (+), subtraction (-), division (/), multiplication (*) and raise to a power (^). Parentheses are recommended to specify the precedence order.

```
set mnemonic = 2 * (7^3)
```

SIMSS-5.2.4.2.3 Set directive using arithmetic and Boolean functions

A container item may also be set to an arithmetic expression using functions from the following table. The following rules apply:

- The function names must be in lower case.
- The parentheses around the function argument are required syntax.
- Arguments to the Boolean functions must be in decimal.
- The Boolean functions perform a bit-wise AND, OR, etc.

Function Description	Set directive example
Arcsine (x)	Set mnemonic = asin (x)
Arccosine (x)	Set mnemonic = acos (x)
Arctangent (x)	Set mnemonic = atan (x)

Function Description	**Set directive example**
Degrees to radians (x)	Set mnemonic = dtor (x)
Radians to degrees (x)	Set mnemonic = rtod (x)
Square root (x)	Set mnemonic = sqrt (x)
Log (x)	Set mnemonic = ln (x)
Exponent (x)	Set mnemonic = exp (x)
Tangent (x)	Set mnemonic = tan (x)
Sine (x)	Set mnemonic = sin (x)
Cosine (x)	Set mnemonic = cos (x)
Sine in radians (x)	Set mnemonic = rsin (x)
Cosine in radians (x)	Set mnemonic = rcos (x)
Arcsine in radians (x)	Set mnemonic = rasin (x)
Arccosine in radians (x)	Set mnemonic = racos (x)
Arctangent in radians (x)	Set mnemonic = ratan (x)
Add x to mnemonic's current value.	Set mnemonic += x
Subtract x from mnemonic's current value	Set mnemonic -= x
Multiply mnemonic's current value by x	Set mnemonic *= x
Divide mnemonic's current value by x	Set mnemonic /= x
Raise mnemonic’s value to the power of x	Set mnemonic ^= x
AND two values together	Set mnemonic = mnemonic and 1
OR two values	Set mnemonic = mnemonic or 7
OR two values and Negate the result	Set mnemonic = mnemonic nor 1
Exclusive OR two values	Set mnemonic = mnemonic xor 7
NAND two values	Set mnemonic = mnemonic nand 15

SIMSS-5.2.4.2.4 Set directive using mnemonics in expressions

Container names may be used instead of numbers in an expression. In the simplest case, a container item may be set to the current value of another container item.

set mnemonicA = mnemonicB

Container items may also be combined together to create expressions.

set mnemonicA = (mnemonicB + mnemonicC)/2

SIMSS-5.2.4.3 Setbuffer Directive

The **setbuffer** directive may be used to change the value of any modifiable buffer in the container for the indicated module. The format of the **setbuffer** directive is

[setbuffer | setbufferle | setbufferandlength] *buffername* <*offset*> [byte | word | dword] <*value*> <*value*> <*value*> …

buffer. If "setbufferandlength" is used, then the size of the buffer is set to the total size of the data values set. The **setbuffer** directive entry line is limited to 100 characters. The names of alterable buffers are provided in the container description for each module. An example of setbuffer usage is:

setbuffer TlmPacket0140 20 byte 10 012 0xA

In this example, three bytes of the buffer TlmPacket0140 beginning at offset 20 are set to 10. If the buffer is successfully set, an event message of the form

"module: values set in buffer *buffername*"

informs the operator of the change. If the buffer name is not recognized or the buffer cannot be modified, an event message of the form

"module: *buffername* not found"

will inform the operator. If unsuccessful, verify that the module indicator is correct, that the buffer name spelling is correct, and that the buffer is not defined as read-only.

SIMSS-5.2.4.4 Get Directive

The **get** directive may be used to display the value of any variable in the container for the indicated module. The format of the **get** directive is

get *name*

If the name matches a variable in the module's container, an event message of the form

"module: *name* = value"

is written to the event message log. If the name doesn't match a container variable, an error event message of the form

"module: *name* not found"

will be produced. If unsuccessful, check the setting of the module indicator and the spelling of the container variable name.

SIMSS-5.2.4.5 Getbuffer Directive

The **getbuffer** directive may be used to observe the value of any modifiable buffer in the indicated module. The format of the **getbuffer** directive is

getbuffer *buffername* <*offset*> [byte | word | dword] <*count*>

where *buffername* is a container buffer name, *offset* is the decimal byte offset to the first item to be retrieved, the keywords "byte", "word", and "dword" indicate the size of data to retrieve at one time, and *count* is the number of items of data to retrieve. If the suffix "le" is added to the typein, the data is retrieved in little-endian (Intel) byte order before being displayed. The **getbuffer** directive entry line is limited to 100 characters. The names of retrievable buffers are provided in the container description for each module. An example of **getbuffer** usage is:

getbuffer TlmPacket0140 20 byte 10

In this example, ten bytes of the buffer TlmPacket0140 beginning at offset 20 are retrieved. If there are no errors, an event message of the form

"module: *buffername* = 14 15 16 17 18 19 1a 1b 1c 1d" informs the operator of the data with values in hexadecimal. If the buffer name is not recognized, an event message of the form

> “module: *buffername* not found”

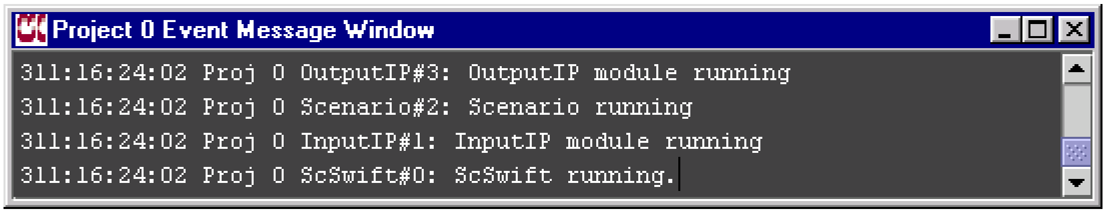
will inform the operator. If unsuccessful, verify that the module indicator is correct, and that the buffer name spelling is correct.

SIMSS-5.3 Project Event Message Window

The Project Event Message window is automatically displayed when a project is added or restored. This window may be resized, minimized or closed. This window defaults to an automatic scrolling mode where the most recent event messages are always displayed. The scroll bar may be used to manually scroll back to display older event messages. The window then stays at the manually scrolled position. To return to automatic scrolling mode, return the scroll bar to the bottom edge of the scroll region.

The project event log window may also be opened and closed by checking and unchecking the “Show Event Log” option from the System menu.

The contents of this window may be filtered by event message type. This window defaults to displaying only information, warning and error level messages. See the section on Event Message Filtering under the project system menu for more information.



```
311:16:24:02 Proj 0 OutputIP#3: OutputIP module running
311:16:24:02 Proj 0 Scenario#2: Scenario running
311:16:24:02 Proj 0 InputIP#1: InputIP module running
311:16:24:02 Proj 0 ScSwift#0: ScSwift running.
```

EOS Ground Station Module (EOSGS)

EOSGS-1.0 Overview

The Earth Observing System (EOS) Ground Station (EOSGS) module is responsible for receiving telemetry, adding an EDOS Service header (ESH) to telemetry packets and transmitting the data as EDOS Data Units (EDUs). The EOSGS module provides a user interface to set EDOS Service header fields. The EOSGS module also receives Command Data Blocks (CDB), validates and strips off the Ground Message Header (GMH), and transmits Command Link Transmission Units (CLTUs).

EOSGS-2.0 Inputs

Ch.	Data expected	Validation performed	Processing performed
1	EDOS Command Data Blocks	Checks GMH fields	Ground Message Header removed, resulting data transmitted through output channel 1
2	Telemetry packets	None	EDOS Service Header added, resulting data transmitted through output channel 2
3	Command Link Control Words (CLCWs)	None	EDOS Service Header added, resulting data transmitted through output channel 3

EOSGS-3.0 Outputs

Ch.	Description
1	Command stream with acquisition sequence and CLTUs
2	EDOS Data Units (EDUs) containing telemetry packets
3	EDOS Data Units (EDUs) containing CLCWs

EOSGS-4.0 Container Items

The EOSGS module accepts operator directives and is capable of receiving directives from a Scenario module. Use the Set and Get directives to access items with a fixed type. Use the SetBuffer and GetBuffer directives on buffer types. Although names in the following tables contain upper and lower case, directive lines are not case-sensitive.

EOSGS-4.1 Counters

Name	Type	Description
GSCmdReceiveCount	Fixed	Number of command blocks received
GSTlmCh1TransmitCount	Fixed	Number of channel 1 blocks sent
GSClcwTransmitCount	Fixed	Number of channel 2 blocks sent

EOSGS-4.2 Buffers

Name	Type	Description
GSCmdBuffer	Buffer	Most recent command block received
GSTlmCh1Buffer	Buffer	Most recent telemetry block sent
GSClcwBuffer	Buffer	Most recent CLCW block sent

EOSGS-4.3 Ground Message Header (GMH) fields

Name	Type	Description
GSCmdEnableHeaderValidation	Fixed	GMH validation flag (1 = enabled, 0 = disabled)
GSCmdExpectedDestination	Fixed	GMH expected destination (1)
GSCmdExpectedMsgType	Fixed	GMH expected message type (3)
GSCmdExpectedSCID	Fixed	GMH expected SCID (0x00)
GSCmdExpectedSource	Fixed	GMH expected source (4)
GSCmdSequenceCount	Fixed	GMH expected sequence count (0)

EOSGS-4.4 EDOS Data Unit (EDU) Header fields

There are two sets of EDU container items, one set for each telemetry channel. Each EDU has an EDOS Service Header (ESH). To access fields from telemetry channel 1, insert "GsTlmch1" for "<ch>" in the names provided in the following table. To access fields from telemetry channel 2, insert "GsClcw" for "<ch>". For example, the container name for the ESH version field of the most recent channel 1 block is "EduHeaderGsTlmCh1Version". The container name for the same field in the most recent channel 2 block is "EduHeaderGsClcwVersion".

Name	Type	Description
EduHeader<ch>Version	Fixed	ESH version field
EduHeader<ch>SDUtype	Fixed	ESH SDU type field
EduHeader<ch>PortId	Fixed	ESH Port Id
EduHeader<ch>SourceDiscontinuityFlag	Fixed	ESH Source VCDU Discontinuity Flag
EduHeader<ch>PlaybackDataFlag	Fixed	ESH Contains Playback Data Flag
EduHeader<ch>RecoveryProcessingFlag	Fixed	ESH Recovery Processing Indicator
EduHeader<ch>TestDataFlag	Fixed	ESH Test Data Indicator
EduHeader<ch>CRCfailureFlag	Fixed	ESH CRC Failure Indicator
EduHeader<ch>PathDiscontinuityFlag	Fixed	ESH Path SDU Discontinuity Flag
EduHeader<ch>PktLengthErrorFlag	Fixed	ESH Packet Length Indicator
EduHeader<ch>PacketFillFlag	Fixed	ESH Packet Fill Indicator
EduHeader<ch>FirstSpare	Fixed	ESH First Spare Field
EduHeader<ch>SCID	Fixed	ESH SCID
EduHeader<ch>VCID	Fixed	ESH VCID
EduHeader<ch>FillLocation	Fixed	ESH Fill Data Location
EduHeader<ch>SecondSpare	Fixed	ESH Second Unused Field

EduHeader<ch>RSstatus	Fixed	ESH RS Validation Status
EduHeader<ch>HeaderErrorCount	Fixed	ESH VCDU Header Error Count
EduHeader<ch>TotalErrorCount	Fixed	ESH Total Error Count

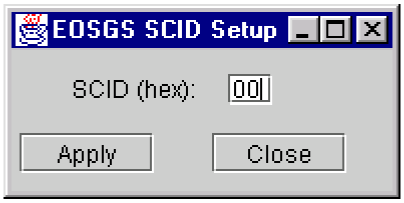
EOSGS-5.0 Displays

To access the displays for this module, first click on the center of the module in the project window. The module pop-up menu will appear.

Module Pop-Up Menu Item	Description
Configuration	Setup expected Spacecraft ID (SCID)
Run-time	Access the Run-time menu for the module
Remove	Remove module from the project
About	Display generic module information

EOSGS-5.1 Configuration

There is one display available at configuration time.



SCID (hex): 00

This display is used to specify the expected Spacecraft Identifier (SCID) field for Ground Message Header validation. This value can be saved and restored as part of a project. Click **Apply** for the entered hexadecimal value to take effect. Use **Close** to dismiss the display.

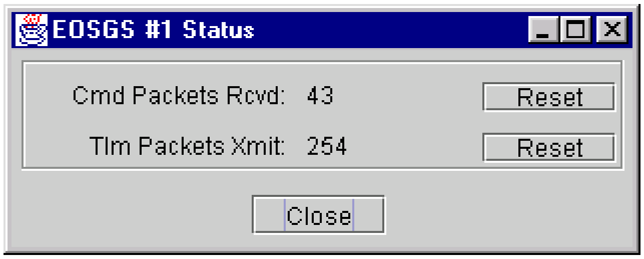
This value will also be used in the SCID fields of outgoing EDOS Service Headers for housekeeping and CLCW telemetry unless overridden. See sections EOSGS-5.2.4 Modify Tlm Header and EOSGS-5.2.5 Modify CLCW Header for more information.

EOSGS-5.2 Run-time

There are six displays available from the Run-time menu.

Run-time Menu Item	Description
Show Status	Show counts of commands and telemetry packets
Show Cmd Packet	Display the command buffer
Show Tlm CH1 Packet	Display the housekeeping telemetry buffer
Modify Tlm Header	Modify header fields in housekeeping telemetry
Modify CLCW Header	Modify command link control word headers
Set/Display GMT	Display time fields for possible modification

EOSGS-5.2.1 Show Status

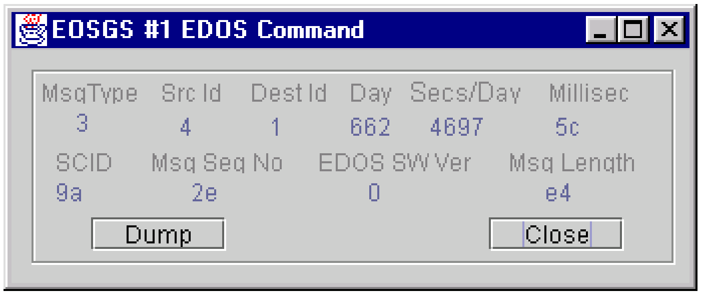


Cmd Packets Rcvd: 43

Tlm Packets Xmit: 254

The status display shows the number of command blocks received and the number of housekeeping telemetry blocks transmitted. The **Reset** button may be used to zero out the counter in the same row.

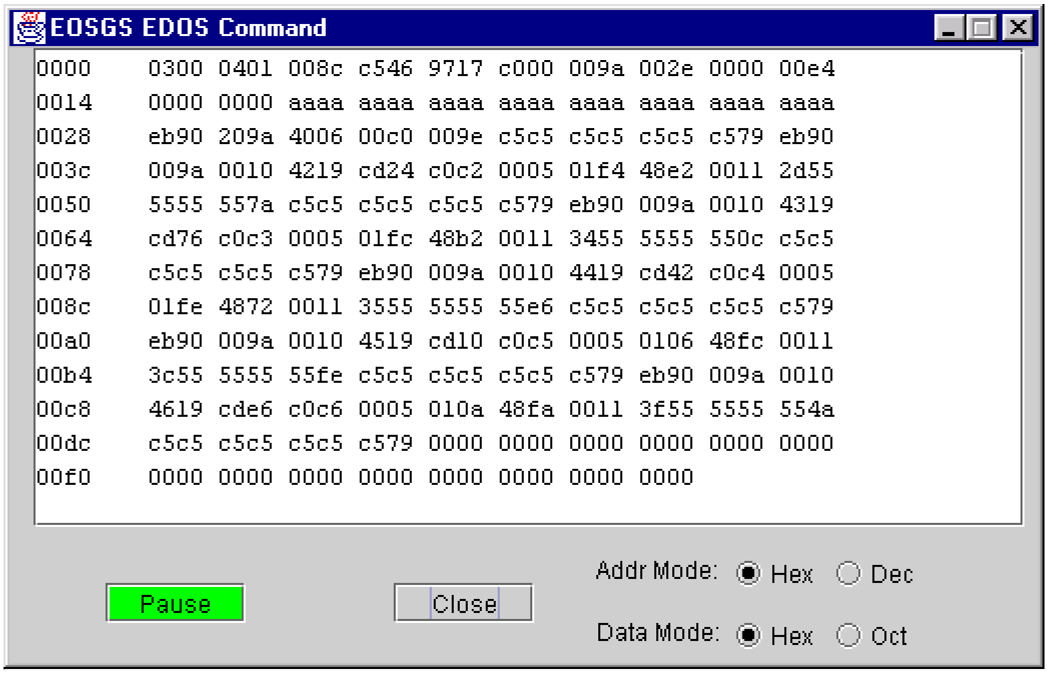
EOSGS-5.2.2 Show Command Packet


EOSGS #1 EDOS Command

MsgType	Src Id	Dest Id	Day	Secs/Day	Millisec
3	4	1	662	4697	5c

SCID	Msg Seq No	EDOS SW Ver	Msg Length
9a	2e	0	e4

The command packet display shows the header and, optionally (by pressing the **Dump** button), the contents of the most recent command block received, including the Ground Message Header.


EOSGS EDOS Command

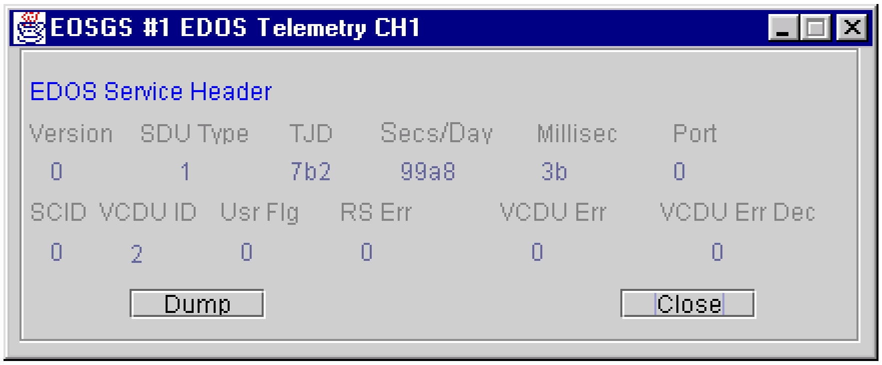
```
0000     0300 0401 008c c546 9717 c000 009a 002e 0000 00e4
0014     0000 0000 aaaa aaaa aaaa aaaa aaaa aaaa aaaa aaaa
0028     eb90 209a 4006 00c0 009e c5c5 c5c5 c5c5 c579 eb90
003c     009a 0010 4219 cd24 c0c2 0005 01f4 48e2 0011 2d55
0050     5555 557a c5c5 c5c5 c5c5 c579 eb90 009a 0010 4319
0064     cd76 c0c3 0005 01fc 48b2 0011 3455 5555 550c c5c5
0078     c5c5 c5c5 c579 eb90 009a 0010 4419 cd42 c0c4 0005
008c     01fe 4872 0011 3555 5555 55e6 c5c5 c5c5 c5c5 c579
00a0     eb90 009a 0010 4519 cd10 c0c5 0005 0106 48fc 0011
00b4     3c55 5555 55fe c5c5 c5c5 c5c5 c579 eb90 009a 0010
00c8     4619 cde6 c0c6 0005 010a 48fa 0011 3f55 5555 554a
00dc     c5c5 c5c5 c5c5 c579 0000 0000 0000 0000 0000 0000
00f0     0000 0000 0000 0000 0000 0000 0000 0000
```

Addr Mode: ● Hex ○ Dec

Data Mode: ● Hex ○ Oct

The address field of the dump display may be toggled between hexadecimal and decimal display formats. The data portion may be toggled between hexadecimal and octal formats.

EOSGS-5.2.3 Show Telemetry Channel 1 Packet

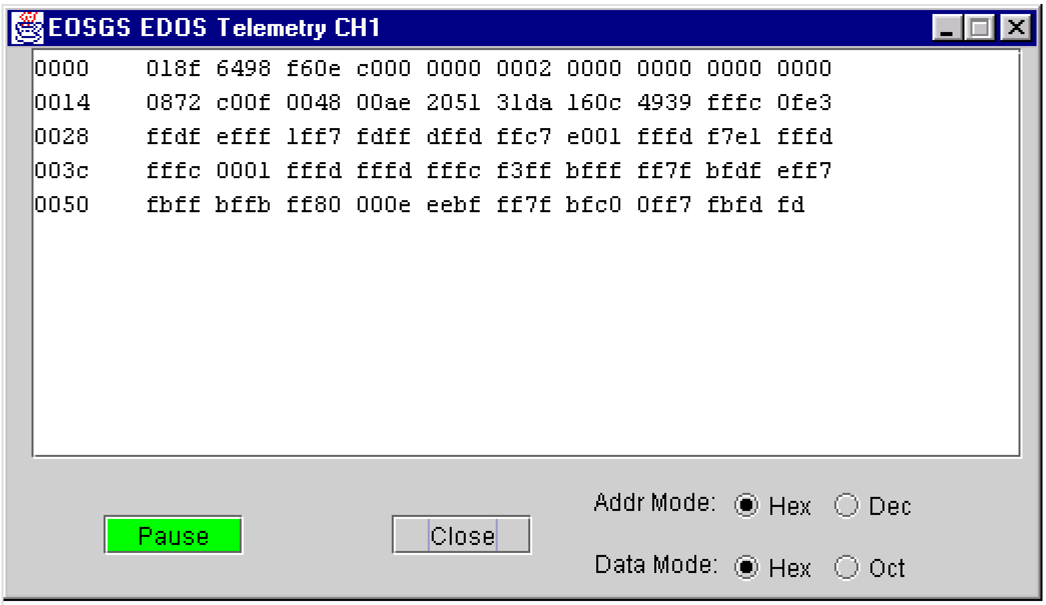

EOSGS #1 EDOS Telemetry CH1

EDOS Service Header

Version	SDU Type	TJD	Secs/Day	Millisec	Port
0	1	7b2	99a8	3b	0

SCID	VCDU ID	Usr Flg	RS Err	VCDU Err	VCDU Err Dec
0	2	0	0	0	0

The telemetry channel 1 packet display shows the header and, optionally (by pressing the **Dump** button), the contents of the most recent telemetry block as transmitted through telemetry channel 1 (module output channel 2).

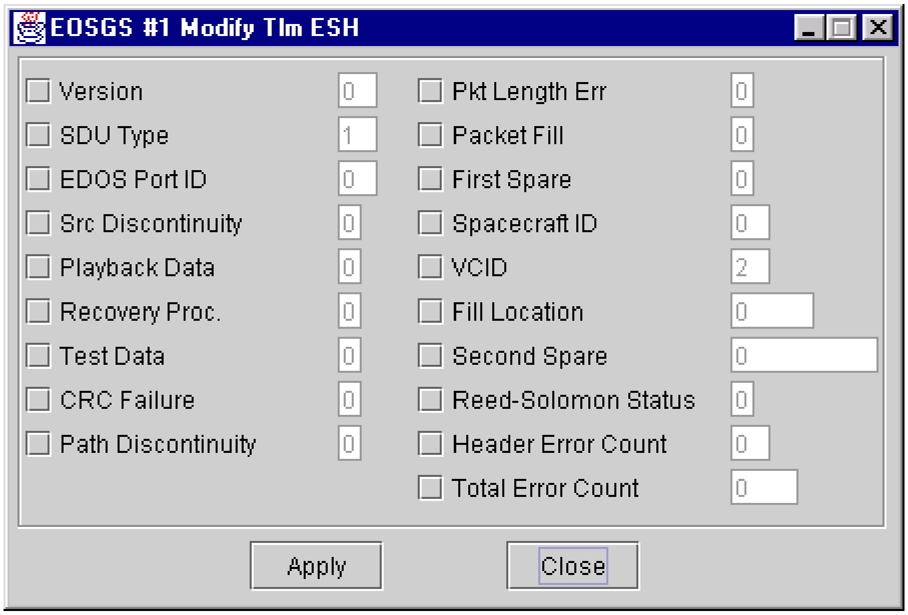


```
0000     018f 6498 f60e c000 0000 0002 0000 0000 0000 0000
0014     0872 c00f 0048 00ae 2051 31da 160c 4939 fffc 0fe3
0028     ffdf efff 1ff7 fdff dffd ffc7 e001 fffd f7e1 fffd
003c     fffc 0001 fffd fffd fffc f3ff bfff ff7f bfdf eff7
0050     fbff bffb ff80 000e eebf ff7f bfc0 0ff7 fbfd fd
```

Addr Mode: ● Hex ○ Dec

Data Mode: ● Hex ○ Oct

EOSGS-5.2.4 Modify Telemetry Header

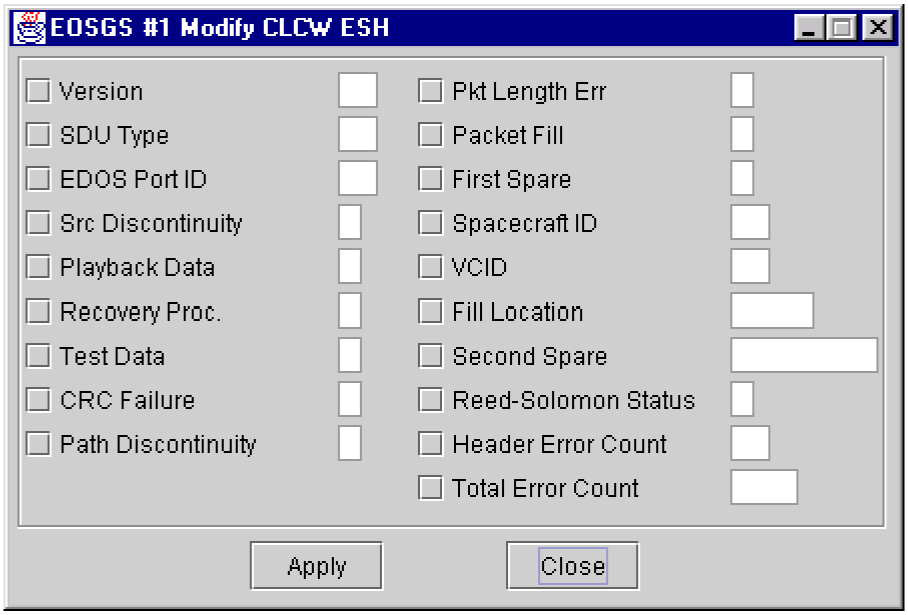


☐ Version	0	☐ Pkt Length Err	0
☐ SDU Type	1	☐ Packet Fill	0
☐ EDOS Port ID	0	☐ First Spare	0
☐ Src Discontinuity	0	☐ Spacecraft ID	0
☐ Playback Data	0	☐ VCID	2
☐ Recovery Proc.	0	☐ Fill Location	0
☐ Test Data	0	☐ Second Spare	0
☐ CRC Failure	0	☐ Reed-Solomon Status	0
☐ Path Discontinuity	0	☐ Header Error Count	0
		☐ Total Error Count	0

The Modify Tlm ESH display allows the operator to change values in the EDOS service header of housekeeping telemetry. Some these changes are in effect for a single packet. Fields marked with an asterisk (*) in the following table are non-sticky. That is, they reset to zero after the transmission of one packet. To change the PB5 Time field, use the Set/Display GMT display.

Modify Tlm ESH Fields	Description
Version	ESH Version Number
SDU Type	SDU Type
EDOS Port ID	EDOS Port ID
Src Discontinuity*	Source VCDU Sequence Counter Discontinuity
Playback Data	VCDU Contains Playback Data
Recovery Proc	Recovery Processing Indicator
Test Data	Test Data Indicator
CRC Failure*	CRC Failure Indicator
Path Discontinuity*	Path SDU Source Seq. Counter Discontinuity
Pkt Length Err*	Packet Length Error
Packet Fill*	Packet Fill Indicator
First Spare	Spare field
Spacecraft ID	Spacecraft identification code
VCID	VCDU Id
Fill Location	Location 1st byte Fill Data For Path SDU
Second Spare	Spare field
Reed-Solomon Status*	RS Error Control Flag
Header Error Count	Source VCDU Header Error Decode Results
Total Error Count	Source VCDU Error Decode Results

EOSGS-5.2.5 Modify CLCW Header



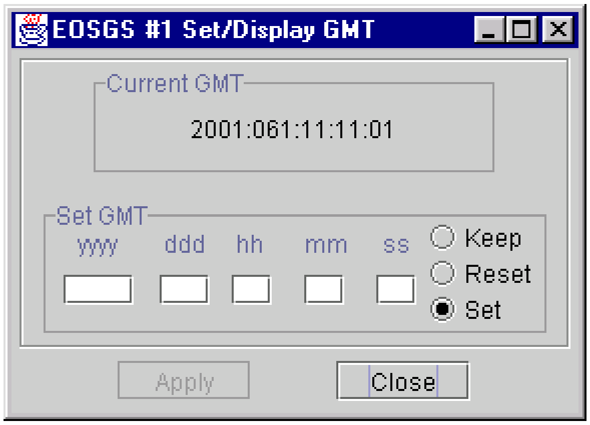
EOSGS #1 Modify CLCW ESH

□ Version	□ Pkt Length Err
□ SDU Type	□ Packet Fill
□ EDOS Port ID	□ First Spare
□ Src Discontinuity	□ Spacecraft ID
□ Playback Data	□ VCID
□ Recovery Proc.	□ Fill Location
□ Test Data	□ Second Spare
□ CRC Failure	□ Reed-Solomon Status
□ Path Discontinuity	□ Header Error Count
	□ Total Error Count

The ModifyCLCW ESH display allows the operator to change values in the EDOS service header of command link control words. Some these changes are in effect for a single packet. Fields marked with an asterisk (*) in the following table are non-sticky. That is, they reset to zero after the transmission of one packet. To change the PB5 Time field, use the Set/Display GMT display.

Modify CLCW ESH Fields	Description
Version	ESH Version Number
SDU Type	SDU Type
EDOS Port ID	EDOS Port ID
Src Discontinuity*	Source VCDU Sequence Counter Discontinuity
Playback Data	VCDU Contains Playback Data
Recovery Proc	Recovery Processing Indicator
Test Data	Test Data Indicator
CRC Failure*	CRC Failure Indicator
Path Discontinuity*	Path SDU Source Seq. Counter Discontinuity
Pkt Length Err*	Packet Length Error
Packet Fill*	Packet Fill Indicator
First Spare	Spare field
Spacecraft ID	Spacecraft identification code
VCID	VCDU Id
Fill Location	Location 1st byte Fill Data For Path SDU
Second Spare	Spare field
Reed-Solomon Status*	RS Error Control Flag
Header Error Count	Source VCDU Header Error Decode Results
Total Error Count	Source VCDU Error Decode Results

EOSGS-5.2.6 Set/Display GMT



Current GMT

2001:061:11:11:01

○ Keep
○ Reset
● Set

This display shows the GMT time that is used in the EDOS Server Header. The user may **Set** the GMT time fields manually, **Keep** the time currently being used, or **Reset** time to the current system time.

Click the **Apply** button for the actions to take effect. Click the **Close** button to dismiss the window with no further action taken.

EOSGS-5.3 About

To display generic information about the EOSGS module, choose the “About” option from the module pop-up menu.

EOSGS-6.0 Special Operating Instructions

There are no special operating instructions for the current release.

Spacecraft Simulation Module for Aura (SCAURA)

SCAURA-1.0 Overview

The functions of the Spacecraft module for Aura (SCAURA) include generation and transmission of formatted telemetry packets, receipt of command link transmission units (CLTUs), and maintenance of GMT and spacecraft time. Spacecraft time is maintained as an offset from the Aura epoch and is inserted into telemetry packets prior to sending them to the Ground Station module. This module is also capable of displaying telemetry packets and received commands under operator direction.

SCAURA-2.0 Inputs

Ch.	Data expected	Validation performed	Processing performed
1	Scenario directives	Same as for operator directives	Directives from a scenario module are processed as if they had come from the user interface directive line.
2	Command stream	CCSDS and Aura specific	Commands parsed and validated per CCSDS standard and Aura extensions. Event messages generated on valid or invalid commands.

SCAURA-3.0 Outputs

The SCAURA module has 2 types of output channels. When the module is using IP mode, only channels 1 and 2 are used. When the module is using Serial mode only channel 3 is used.

Ch.	Description
1	IP mode Housekeeping telemetry packets
2	IP mode Command Link Control Words (CLCWs)
3	Serial mode telemetry transmission

SCAURA-4.0 Container Items

This module accepts operator directives and is capable of receiving directives from a Scenario module. Use the **Set** and **Get** directives to access items with a fixed or string type. Use the **SetBuffer** and **GetBuffer** directives on buffer types. Although names in the following tables contain upper and lower case, directive lines are not case sensitive.

SCAURA-4.1 Telemetry Container Items

The container items for telemetry processing fall into four major groups: control and status of telemetry streams, telemetry points, packet buffers, and VCDU buffers.

SCAURA-4.1.1 Telemetry Status and Control Container Items

Name	Type	Description
CLCWFrameCount	Fixed	CLCW channel transmitted count
CLCWPkt	Fixed	CLCW packet buffer
TlmChannel1Enabled	Fixed	Channel 1 enabled flag (0=disabled, 1=enabled) (default is disabled)
TlmChannel1FrameCount	Fixed	Channel 1 frames transmitted count
TlmCUCpfield	Fixed	P-field for CUC time in packet header
TlmCUCpfieldextension	Fixed	P-field extension for CUC time in packet header
TlmEnabledCH1	Fixed	Flag to start/stop serial channel 1
TlmEnabledCH2	Fixed	Flag to start/stop serial channel 2
TlmInterval3	Fixed	Interval for automatic sending of VCDUs on physical channel 3
TlmOutputFormat	Fixed	Flag to change output format (0=IP, 1=serial)
TlmPktDropCnt	Fixed	Used to check if any VCDU packets were dropped
TransmitCLCW	Fixed	CLCW enabled flag (0=disabled, 1=enabled) (default is disabled)
CLCWInterval	Fixed	Interval between CLCW packet transmissions in seconds
CLCWTransmitLinked	Fixed	CLCW transmission linked to H/K transmission (0=no, 1=yes) (default is linked)

SCAURA-4.1.2 Telemetry Point Container Items

Each telemetry point read from the database is defined in the container four ways. The raw value is identified by the telemetry mnemonic name or by "TLM#" followed by the parameter identification number in decimal digits. The value in engineering units is identified by one of the raw names with "__EU" (two underscores) added to the end. For example, the raw value for telemetry point COM_BR_TXB (parm_id=3525) may be accessed with the name "COM_BR_TXB" or "TLM#3525". Its engineering value is identified by "COM_BR_TXB__EU" or "TLM#3525__EU". The value of any telemetry point can be requested using a **get** directive. The value of any telemetry point can be changed using the **set** directive. Whenever a telemetry point's value is changed, its raw and engineering values are updated using the conversion equations from the database. Set directives may also be used within scenario files.

Note: The raw values of telemetry points are stored as integers and the engineering unit values are stored as floating point values. When floating point values are assigned to raw values, the fractional portion is dropped.

SCAURA-4.1.3 Telemetry Packet Container Items

For each valid telemetry packet defined in the database, container items are created. Where “<APID>” appears in the following table, substitute the packet's APID number in

4 decimal digits. For example, the container buffer for APID 397 is named “TlmPacket0397”. For packets that have secondary keys, substitute the APID in 4 decimal digits plus the secondary key in 4 decimal digits for the "<APID>". For example, the container buffer for APID 114 with secondary key of 55, is named "TlmPacket01140055".

Name	Type	Description
TlmPacket<APID>	Buffer	Buffer for specified APID’s telemetry packet.
TlmPacket<APID>SequenceCount	Fixed	Sequence count for APID’s packet header
TlmPacket<APID>SequenceFlag	Fixed	Sequence flag for APID’s packet header
TlmPacket<APID>APID	Fixed	APID for packet header
TlmPacket<APID>SecondaryHeaderFlag	Fixed	Secondary header flag for APID’s packet header
TlmPacket<APID>DataLength	Fixed	Data length for APID’s packet header
TlmPacket<APID>Version	Fixed	Version for APID’s packet header
TlmPacket<APID>Type	Fixed	Type for APID’s packet header
TlmPacket<APID>Enabled	Fixed	Enabled flag for packet (0=no, 1=yes) (default is not enabled)
TlmPacket<APID>Interval	Fixed	Time between transmissions in seconds

SCAURA-4.1.4 Segmented Telemetry Container Items

In order to implement segmented telemetry packets, a file named **SegmentedApids.txt**, containing APID numbers, one per line, must be in the properties folder. This causes a segment data buffer and some associated variables to be created in addition to the telemetry items shown in the previous section. In the the table below, substitute the appropriate segmented APID in 4 decimal digits for the <APID> in the Name. Scenario files should be used to store data into these segmented buffer items and control their tranmission. A sample scenario file to initialize and transmit APID 1715 is **TES Boilerplate 3 segments.txt**, which should be in the default scenario directory.

Name	Type	Description
FPCResponseData<APID>	Buffer 5000 bytes	Segmented telemetry buffer that is sized to hold all segment data. Contents should be configured using **setBuffer** directives.
FPCResponseLength<APID>	Fixed	Variable that specifies the length of the above segment data buffer.
FPC_ID<APID>	Fixed	Variable used to set the FPC cooler Id, a 2 byte field that follows the secondary header. Can be 1 or 2 for TES APIDs.

SCAURA-4.1.5 Telemetry Dump Packet Container Items

The dump packets are not generated from the database. In order to simulate a dump, the following dump parameters must be set. **As soon as the dump word count is given**, the dump packets will be formatted and transmitted until all of the specified words have been dumped. If dump parameters are being set via a scenario script, the dump word count must be set last. The dump word buffer may be set by a series of setBuffer directives, if not copied from a command load. The dump header information may be set using directives or the “Initiate Dump Packets…” selection from the Telemetry Menu.

Command load processing optionally copies the first segment of data words in a command load to the telemetry dump data buffer. As long as successive command loads contain consecutive memory locations, data words will be copied. Event messages produced during command load processing will provide information for starting a telemetry dump of the loaded memory locations.

Name	**Type**	**Description**
CmdLoadCopyToDump	Fixed	Automatic copying of command load buffer to telemetry dump buffer enabled flag (0=disabled, 1=enabled) (default is enabled)
CmdLoadValidChecksum	Fixed	Flag for validating command load buffer's checksum (0=disabled, 1=enabled) (default is enabled)
DumpMemoryAddress	Fixed	Memory address expected for next consecutive command load
DumpMemoryPosition	Fixed	Current byte position for automatic copy into the telemetry dump data buffer
CmdSpaceLoadBuffer	Buffer	Spacecraft memory load buffer (1024 bytes)
TlmDumpAPID	Fixed	Dump packet APID
TlmDumpInterval	Fixed	Interval in milliseconds between dump packets: 10000 = 10 seconds, defaults to 1000 (1 second)
TlmDumpPacketSize	Fixed	Size of dump packet (default is 256)
TlmDumpSecondaryHeader	Fixed	0 = no secondary header 1 = spacecraft-typesecondary header with CUC (defaults to no secondary header)
TlmDumpStartAddress	Fixed	Starting memory dump address
TlmDumpWordCount	Fixed	Number of 16 bit words to dump
SCMemory	Buffer	6000 byte telemetry dump data buffer

SCAURA-4.1.6 Telemetry VCDU Container Items

For the transmission of serial mode telemetry, certain VCDU Primary Header field values are defined in container items so that they may be easily changed. These are listed in the following table. All Aura telemetry is normally transmitted on virtual channel 2. Modification of the Fill VCDU (Virtual Channel 63) is not possible. To set the replay flag, for example, enter the directive ‘set vcdu2replay 1’.

Name	Type	Size	Description
vcdu2version	Fixed	2 bits	Version Number
vcdu2scid	Fixed	8 bits	Spacecraft ID
vcdu2vcid	Fixed	6 bits	Virtual Channel ID
vcdu2replay	Fixed	1 bit	Replay Flag
vcdu2reserved	Fixed	7 bits	Reserved Flags

It is also possible to inject data values into VCDU headers that will appear as intermittent errors in the data stream. There are 50 internal buffers where VCDUs are constructed. To inject an error that will appear once in every 50 VCDUs, enter a set directive using one of the following container names as the argument. Where "<buffer>" appears in the following table, substitute a buffer number between zero and 49. For example, the container item for buffer 11's Primary Header Version field is named "VCDUVERSION11".

Name	Type	Size	Description
VCDUVERSION<buffer>	Fixed	2 bits	Version Number
VCDUSCID<buffer>	Fixed	8 bits	Spacecraft ID
VCDUVCID<buffer>	Fixed	6 bits	Virtual Channel ID
VCDUREPLAY<buffer>	Fixed	1 bit	Replay Flag
VCDURESERVED<buffer>	Fixed	7 bits	Reserved Flags

SCAURA-4.2 Command Container Items

The container items for command processing fall into several groups: mission specific items, validation/debug flags, CLCW fields, processing buffers, command triggering scenarios, command sub-mnemonics and counts. Initial values for some items are shown in parentheses following the description.

SCAURA-4.2.1 Mission Specific Container Items

The initial values for these items are shown in parentheses. Values may be changed using **set** or **setBuffer** directives from the directive line or from a scenario file.

Name	Type	Description
CmdCltuCodeblockSize	Fixed	Codeblock size in bytes (8)
CmdCltuExpectedStartSequence	Buffer	Expected CLTU start sequence buffer (2 bytes) ($EB90_{16}$)
CmdCltuExpectedTailSequence	Buffer	Expected CLTU tail sequence buffer (8 bytes) (C5C5 C5C5 C5C5 $C579_{16}$)
CmdCriticalTieAVCID	Fixed	Virtual Channel ID (VCID) for TIE-A critical commands (16)
CmdCriticalTieBVCID	Fixed	Virtual Channel ID for TIE-B critical commands (17)
CmdSCID	Fixed	Spacecraft Identifier (CC_{16})
CmdSlidingWindowSize	Fixed	FARM-1 Sliding Window Size (180)

SCAURA-4.2.2 Command Validation/Debug Flags

Name	Type	Description
CmdEnabled	Fixed	Command processing enabled flag. (0=disabled, 1=enabled) (default is enabled)
CodeblockValidation	Fixed	Command codeblock parity validation enabled flag. (0=disabled, 1=enabled) (default is enabled)
FrameValidation	Fixed	Command transfer frame header validation enabled flag. (0=disabled, 1=enabled) (default is enabled)
FARMValidation	Fixed	Command Frame Acceptance and Reporting Mechanism (FARM) validation enabled flag. (0=disabled, 1=enabled) (default is enabled)
PacketValidation	Fixed	Command packet validation enabled flag. (0=disabled, 1=enabled) (default is enabled)
CmdDebugEnabled	Fixed	Command master debug messages flag. (0=disabled, 1=enabled) When enabled, additional event messages are generated throughout command ingest processing. (default is disabled)
CmdDebugCLTU	Fixed	CLTU debug messages flag (0=disabled, 1=enabled) When enabled, CLTU debug event messages are produced. This is a subset of the CmdDebugEnabled messages. (default is disabled)
CmdDebugCodeblock	Fixed	Codeblock debug messages flag (0=disabled, 1=enabled) When enabled, codeblock debug event messages are produced. This is a subset of the CmdDebugEnabled messages. (default is disabled)
CmdDebugFrame	Fixed	Frame debug messages flag (0=disabled, 1=enabled) When enabled, frame debug event messages are produced. This is a subset of the CmdDebugEnabled messages. (default is disabled)
CmdDebugSpacePacket	Fixed	Debug spacecraft packet messages flag (0=disabled, 1=enabled) When enabled, debug event messages are produced for each spacecraft channel packet. This is a subset of the CmdDebugEnabled messages. (default is disabled)

CmdDebugInstrPacket	Fixed	Debug instrument packet messages flag (0=disabled, 1=enabled) When enabled, debug event messages are produced for each spacecraft channel packet. This is a subset of the CmdDebugEnabled messages. (default is disabled)

SCAURA-4.2.3 Command Link Control Word Fields

Name	Type	Description
CmdSpaceClcwCWT	Fixed	Spacecraft CLCW Control Word Type
CmdSpaceClcwVersion	Fixed	Spacecraft CLCW Version
CmdSpaceClcwStatus	Fixed	Spacecraft CLCW Status
CmdSpaceClcwCOP	Fixed	Spacecraft CLCW COP In Effect (1)
CmdSpaceClcwVCID	Fixed	Spacecraft CLCW VCID (0)
CmdSpaceClcwSpare1	Fixed	Spacecraft CLCW Spare field 1
CmdSpaceClcwNoRFAvail	Fixed	Spacecraft CLCW No RF Avail Flag
CmdSpaceClcwNoBitLock	Fixed	Spacecraft CLCW No Bit Lock Flag
CmdSpaceClcwLockout	Fixed	Spacecraft CLCW Lockout Flag
CmdSpaceClcwWait	Fixed	Spacecraft CLCW Wait Flag
CmdSpaceClcwRetransmit	Fixed	Spacecraft CLCW Retransmit Flag
CmdSpaceClcwFarmCount	Fixed	Spacecraft CLCW Farm-B Counter
CmdSpaceClcwSpare2	Fixed	Spacecraft CLCW Spare field 2
CmdSpaceClcwReport	Fixed	Spacecraft CLCW Report Value
CmdInstrClcwCWT	Fixed	Instrument CLCW Control Word Type
CmdInstrClcwVersion	Fixed	Instrument CLCW Version
CmdInstrClcwStatus	Fixed	Instrument CLCW Status
CmdInstrClcwCOP	Fixed	Instrument CLCW COP In Effect (1)
CmdInstrClcwVCID	Fixed	Instrument CLCW VCID (1)
CmdInstrClcwSpare1	Fixed	Instrument CLCW Spare field 1
CmdInstrClcwNoRFAvail	Fixed	Instrument CLCW No RF Avail Flag
CmdInstrClcwNoBitLock	Fixed	Instrument CLCW No Bit Lock Flag
CmdInstrClcwLockout	Fixed	Instrument CLCW Lockout Flag
CmdInstrClcwWait	Fixed	Instrument CLCW Wait Flag
CmdInstrClcwRetransmit	Fixed	Instrument CLCW Retransmit Flag
CmdInstrClcwFarmCount	Fixed	Instrument CLCW Farm-B Counter
CmdInstrClcwSpare2	Fixed	Instrument CLCW Spare field 2
CmdInstrClcwReport	Fixed	Instrument CLCW Report Value

SCAURA-4.2.4 Command Container Buffers

Name	Type	Description
CmdPolyRemainderTbl	Buffer	Polynomial remainder table for parity calculation. (256 bytes)
CmdSpaceCLCW	Buffer	Spacecraft virtual channel CLCW buffer (4 bytes)
CmdInstrCLCW	Buffer	Instrument virtual channel CLCW buffer (4 bytes)

CmdSpacePkt	Buffer	Spacecraft command packet buffer (128 bytes)
CmdInstrPkt	Buffer	Instrument command packet buffer (128 bytes)
CmdFrameBuffer	Buffer	Command transfer frame buffer (256 bytes)
CmdCLTU	Buffer	Command link transmission unit buffer (6000 bytes)
CmdCodeblock	Buffer	Compressed codeblock buffer (holds codeblock bytes without parity bytes) (6000 bytes)

SCAURA-4.2.5 Command Triggering Scenarios Container Items

Name	Type	Description
ScenarioPath	String	Full pathname of directory where the command triggering scenario definition file can be found. Path should end with slash character.
ReadScenFile	Fixed	Whenever a set directive accesses this flag, the command triggering scenario definition file is re-read, replacing the previous definitions.

SCAURA-4.2.6 Command Submnemonic Container Items

For each variable command in the database, container items are created for each sub-mnemonic field. The format for the names of these items is subname#parm_id. For example, the value received in the ANGLE sub-mnemonic for the command with PARM_ID 32081 will be stored in "ANGLE#32081". During run-time as variable commands are identified in the database, event messages display each command's mnemonic name, its parameter ID in decimal digits, and any sub-mnemonic names and values. At the same time that the event message is being created, the received sub-mnemonic values are stored into their associated container items.

SCAURA-4.2.7 Command Counter Container Items

Name	Type	Description
TIE_A_Active	Fixed	Flag indicates whether TIE A or TIE B is online for the purpose of counting codeblocks (0=TIE B, 1=TIE A) (default is TIE_A)
CmdCounter0	String	Name of telemetry point counter for TIE A valid codeblocks
CmdCounter1	String	Name of telemetry point counter for TIE B valid codeblocks
CmdCounter2	String	Name of telemetry point counter for VCID 16 valid critical commands
CmdCounter3	String	Name of telemetry point counter for VCID 17 valid critical command
CmdCounter4	String	Name of telemetry point counter for valid ISC commands
CmdCounter5	String	Name of telemetry point counter for valid PC commands

CmdCounter6	String	Name of telemetry point counter for valid GNCC commands
CmdCounter7	String	Name of telemetry point counter for valid CTC online commands
CmdCounter8	String	Name of telemetry point counter for valid CTC offline commands
CmdCounter9	String	Name of telemetry point counter for invalid ISC commands
CmdCounter10	String	Name of telemetry point counter for invalid PC commands
CmdCounter11	String	Name of telemetry point counter for invalid GNCC commands
CmdCounter12	String	Name of telemetry point counter for invalid CTC online commands
CmdCounter13	String	Name of telemetry point counter for invalid CTC offline commands
CmdTotalCLTUs	Fixed	Count of all CLTUs received
CmdValidCLTUs	Fixed	Count of valid CLTUs
CmdRejectCLTUs	Fixed	Count of invalid CLTUs
CmdTotalCodeblocks	Fixed	Count of all Codeblocks
CmdValidCodeblocks	Fixed	Count of valid Codeblocks
CmdRejectCodeblocks	Fixed	Count of invalid Codeblocks
CmdTotalTransferFrames	Fixed	Count of all Transfer Frames
CmdValidTransferFrames	Fixed	Count of valid Transfer Frames
CmdErrorTransferFrames	Fixed	Count of invalid Transfer Frames
CmdADFrames	Fixed	Count of Type AD Transfer Frames
CmdACFrames	Fixed	Count of Type AC Transfer Frames
CmdBCFrames	Fixed	Count of Type BC Transfer Frames
CmdBDFrames	Fixed	Count of Type BD Transfer Frames
CmdTotalSpacePkts	Fixed	Count of all Spacecraft Packets
CmdValidSpacePkts	Fixed	Count of valid Spacecraft Packets
CmdErrorSpacePkts	Fixed	Count of invalid Spacecraft Packets
CmdTotalInstrPkts	Fixed	Count of all Instrument Packets
CmdValidInstrPkts	Fixed	Count of valid Instrument Packets
CmdErrorInstrPkts	Fixed	Count of invalid Instrument Packets
CmdIgnoredCLTUs	Fixed	Number of CLTUs ignored while command processing is disabled.

SCAURA-4.3 Database Container Items

Name	**Type**	**Description**
DatabaseCmdConnected	Fixed	Command processing connected flag (0=not connected, 1=connected)
DatabaseCmdId	Fixed	Command identification flag (0=off, 1=on). When this flag is on, command identification using the

		database occurs with each received command. If command processing is too slow, this flag should be set to off.
DatabaseDebug	Fixed	Debug flag (0=disabled, 1=enabled). When enabled, database debug messages are written to the server window. (default is disabled)
DatabaseTlmConnected	Fixed	Telemetry initialization connected flag (0=not connected, 1=connected)
DatabaseVersionFound	Fixed	Version found flag (0= not found, 1=found)
TelemetryLoaded	Fixed	Telemetry loaded from database flag (0=not loaded, 1=loaded)

SCAURA-5.0 Displays

To access the displays for this module, first click on the center of the SCAURA module in the project window. The following items will appear in a pop-up menu.

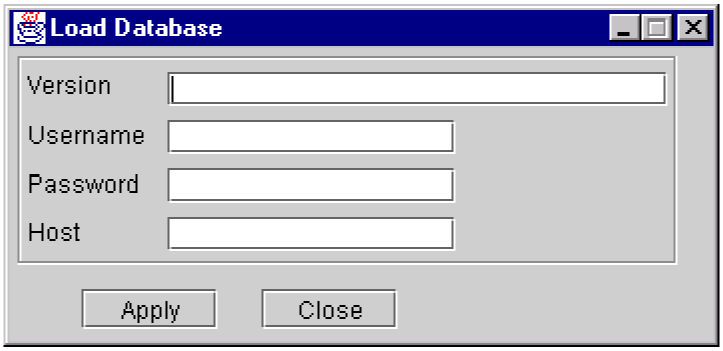
Module Pop-Up Menu Item	Description
Configure	Access the configuration menu for the module
Run-time	Access the Run-time menu for the module
Remove	Remove module from the project
About	Display generic module information

SCAURA-5.1 Configuration Menu

The configuration menu for the SCAURA module contains two items.

Configuration Menu Item	Description
Load Database	Load the database information
Select Simulation Mode	Select IP or Serial Mode

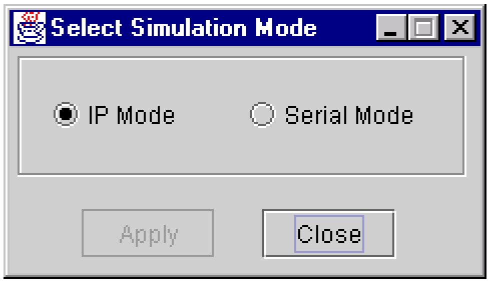
SCAURA-5.1.1 Load Database



This display specifies the database version to use to initialize telemetry information in the simulator. During run-time, this database will also be accessed to identify received commands and generate event messages with command mnemonics. The Username and Password fields must be entered. When the **Apply** button is used, an attempt is made to access and read from the database. Information about the success or failure of the database connection and telemetry initialization is reported to the event message log.

Load Database Field	Description
Version	Optionally specifies a database version. If this field is left blank, the most recently loaded database will be used.
Username	Specifies the user account
Password	Specifies the password for the user account
Host	Optionally specifies a remote database host system. Both the local and remote systems must be properly configured to support remote database access. Refer to delivery package special operating instructions for more information. When this field is left blank, the local database is accessed.

SCAURA-5.1.2 Select Simulation Mode



This display is used to set the Simulation Mode. Depending on the simulation mode, different output interface modules must be used in the project’s design. Clicking the **Apply** button causes the action defined by the fields to take effect. Clicking the **Close** button closes the window with no additional action taken.

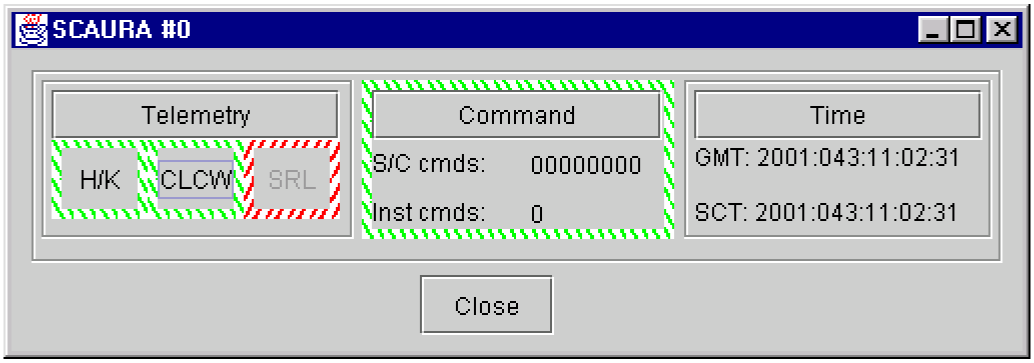
Mode Field	Description
IP Mode	The module will use IP protocol for the output channels. See the IP module chapter for configuration of the Output IP modules.
Serial Mode	The module will use a serial output channel. See the Serial module chapter for configuration of the Serial Output module.

SCAURA-5.2 Run-time Menu

The Run-time menu for the SCAURA module contains the following three items.

Run-time Menu Item	Description
Control	Request the module's main display
Resume	Resume the module's execution
Pause	Pause the module's execution

SCAURA-5.2.1 Main Display



S/C cmds: 00000000
Inst cmds: 0
GMT: 2001:043:11:02:31
SCT: 2001:043:11:02:31

This is the main display for the SCAURA module, providing basic information about its current state, including the current GMT and spacecraft times, the number of valid spacecraft and instrument commands received, and the enabled/disabled status of command receipt and telemetry transmission. The latter are indicated by colored diagonal stripes around the appropriate box: the entire command box for command status and the telemetry channel boxes for each telemetry channel. Red stripes angling up and to the right indicate disabled, while green stripes angling down and to the right indicate enabled. When the module is using IP mode, the telemetry region buttons **H/K** and **CLCW** are available and the **SRL** button is unavailable. When the module is using serial mode, only the **SRL** button is available. Clicking on the **Telemetry**, **Command**, and **Time** buttons in the main window provides access to subordinate displays.

SCAURA-5.2.1.1 Telemetry Menus

There are four buttons in the telemetry area of the main display: **Telemetry**, **H/K**, **CLCW** and **SRL**. Clicking the **Telemetry** button brings up a menu with the following items.

Telemetry Menu Item	Description
Modify Packet…	Modify packet data area on byte basis
Display Packet…	Display packet header and contents
Display Status	Display status of the telemetry channels
Control Packet…	Change the frequency of the packet transmission
Initiate Dump…	Set parameters to initiate dump packets
Display/Set Container Items…	Display and optionally modify container items
Display Container Buffer…	Display any container buffer
Modify CUC…	Modify the CCSDS Unsegmented Time Code
APID Status	Display status of database defined packets

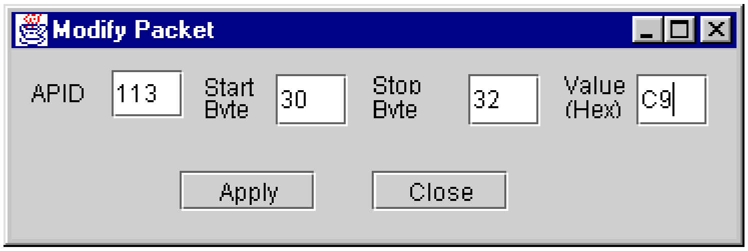
Clicking the **H/K**, **CLCW** or **SRL** channel buttons brings up a menu with the following choices.

Channel Menu Item	Description
Start	Enable telemetry transmission on the channel
Stop	Disable telemetry transmission on the channel

Note: The **H/K** and **CLCW** buttons have been linked together to simplify operations. Whenever the **H/K** button is used to start or stop housekeeping telemetry transmission, the same action will be applied to the CLCW channel. The **CLCW** button may be used to start or stop the CLCW channel independently as well. If decoupling of these two buttons is desired, enter the following directive on the directive line or in a scenario file.

Set ClcwTransmitLinked 0

SCAURA-5.2.1.1.1 Modify Packet Display



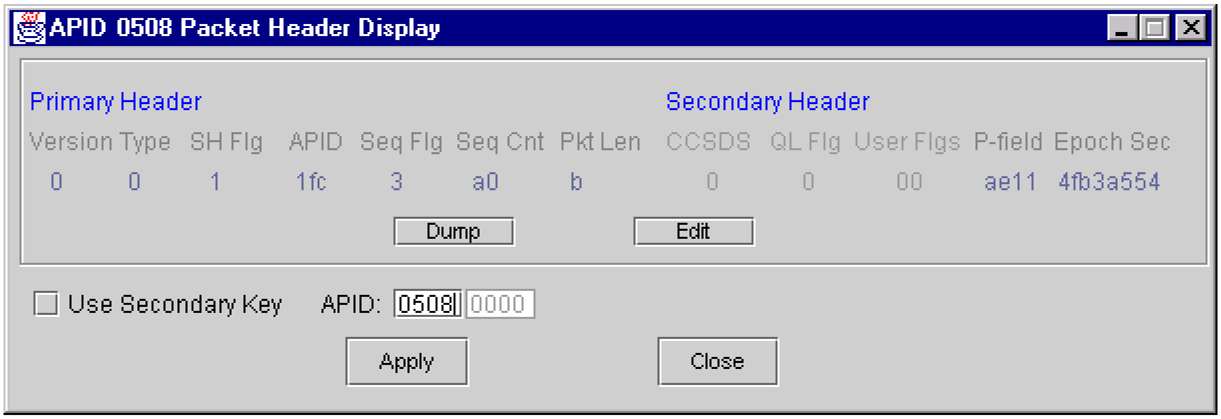
Modify Packet

APID 113 Start Byte 30 Stop Byte 32 Value (Hex) C9

This display allows the user to modify the data area of the packet on a byte or repeating byte basis. All of the display fields must be entered.

Data Entry Field	Description
APID	Application ID of the packet to be modified
Start Byte	First byte to change
Stop Byte	Last byte to change
Value (Hex)	Value to change byte(s) to, in hexadecimal

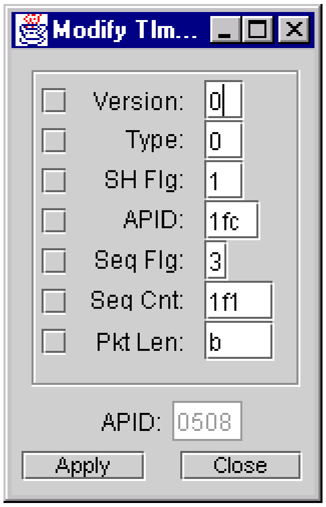
SCAURA-5.2.1.1.2 Telemetry Packet Display


APID 0508 Packet Header Display

Primary Header							Secondary Header				
Version	Type	SH Flg	APID	Seq Flg	Seq Cnt	Pkt Len	CCSDS	QL Flg	User Flgs	P-field	Epoch Sec
0	0	1	1fc	3	a0	b	0	0	00	ae11	4fb3a554

□ Use Secondary Key APID: 0508 0000

The telemetry packet display shows the header and, optionally (by pressing the **Dump** button), the contents of the most recent telemetry packet sent with the given APID. The user must enter a valid value in the APID field and then click the **Apply** button before any other values will be reported. If there are multiple formats for an APID, click the "Use Secondary Key" box and enter the secondary key value in the space next to the APID. Clicking the **Dump** button will bring up a standard dump display with the contents of the packet. Clicking the **Edit** button will bring up the Modify Tlm Packet Header Display shown below. Clicking the **Close** button closes the window with no additional action taken.


Modify Tlm...

□ Version: 0
□ Type: 0
□ SH Flg: 1
□ APID: 1fc
□ Seq Flg: 3
□ Seq Cnt: 1f1
□ Pkt Len: b

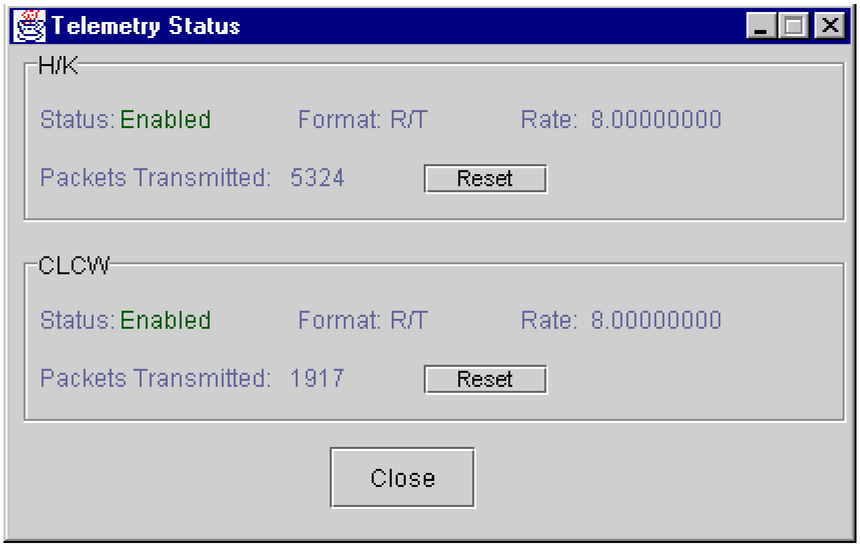
APID: 0508

The Modify Tlm Packet Header display allows the user to modify fields in the packet header. Click on an enable box to the left of the field to enable modification of a specific header field. Clicking the **Apply** button puts the settings into effect. Clicking the **Close** button closes the display with no additional action taken.

The purpose of the Modify Tlm Packet Header display is to inject data values into transmitted packets that could be perceived as erroneous by the receiving system. Changing the APID and Pkt Len fields does not change the name or length of the buffer, only the data in the header fields.

Data Entry Field	Description
Version	CCSDS packet version number
Type	CCSDS packet type
SH Flg	Secondary header flag
APID	Application identifier (packet number)
Seq Flg	Sequence flag
Seq Cnt	Packet sequence counter
Pkt Len	Packet length for data zone (seven less than actual length)

SCAURA-5.2.1.1.3 Telemetry Status Display



H/K

Status: Enabled Format: R/T Rate: 8.00000000

Packets Transmitted: 5324

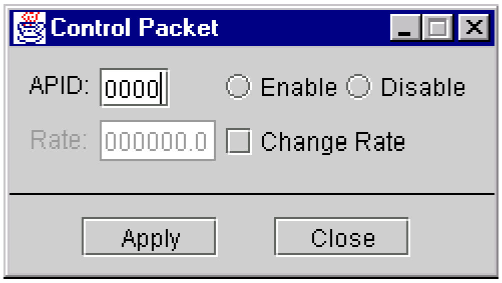
CLCW

Status: Enabled Format: R/T Rate: 8.00000000

Packets Transmitted: 1917

The telemetry status display shows the enabled or disabled status and the number of packets transmitted for each of the two telemetry channels. Clicking the **Reset** button will reset the count of packets transmitted for that channel. Clicking the **Close** button closes this window with no additional action taken.

SCAURA-5.2.1.1.4 Control Packet Display



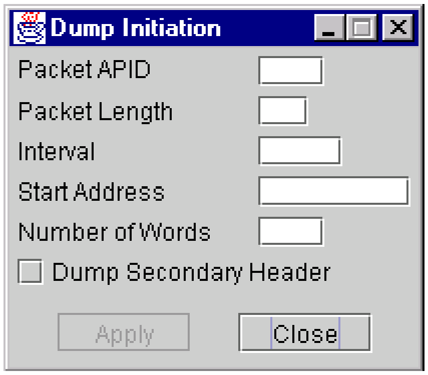
APID: 0000 ○ Enable ○ Disable

Rate: 000000.0 □ Change Rate

This display allows the user to control whether or not a telemetry packet is generated and optionally change the rate at which it is transmitted. Clicking the **Apply** button puts the settings into effect. Clicking the **Close** button exits the display.

Data Entry Field	**Description**
APID	Application ID of the packet to be modified
Enable	If flag is set, packet is formatted and sent
Disable	If flag is set, the packet is not transmitted
Change Rate	This box must be checked in order to change the rate field
Rate	The interval in seconds and fractions of seconds between successive packet transmissions.

SCAURA-5.2.1.1.5 Initiate Dump Packets


Dump Initiation

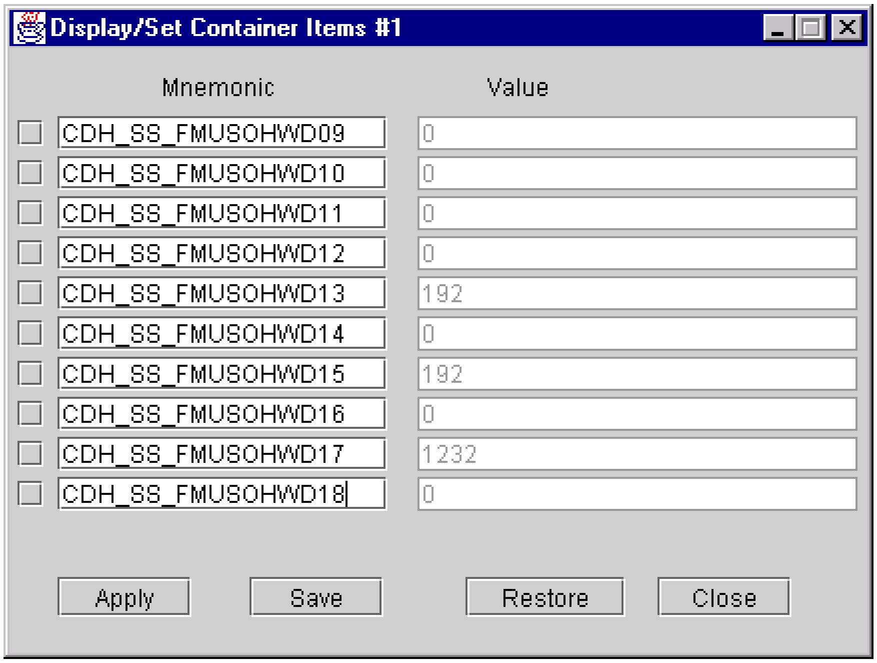
Packet APID	
Packet Length	
Interval	
Start Address	
Number of Words	

□ Dump Secondary Header

This display allows the operator to specify dump packet header values and initiate the sending of dump packets. The dump data words are kept in the container buffer SCMemory. This buffer may be modified using the setBuffer directive on the directive line or within a scenario file. Only one dump may be specified at a time. See Paragraph SCAURA-4.1.5, Telemetry Dump Packet Container Items, for more information.

Data Entry Field	Description
Packet APID	Application ID to store in dump packet header
Packet Length	Maximum size of dump packet. The last packet may be smaller.
Interval	Time between dump packet transmissions, in milliseconds
Start Address	Starting memory address
Number of Words	Number of words to be dumped
Dump Secondary Header	If this box is checked, the dump packets will have a spacecraft-format secondary header containing a CCSDS unsegmented time code. If this box is not checked, the dump packets will not have a secondary header.

SCAURA-5.2.1.1.6 Display/Set Container Items


Display/Set Container Items #1

	Mnemonic	Value
□	CDH_SS_FMUSOHWD09	0
□	CDH_SS_FMUSOHWD10	0
□	CDH_SS_FMUSOHWD11	0
□	CDH_SS_FMUSOHWD12	0
□	CDH_SS_FMUSOHWD13	192
□	CDH_SS_FMUSOHWD14	0
□	CDH_SS_FMUSOHWD15	192
□	CDH_SS_FMUSOHWD16	0
□	CDH_SS_FMUSOHWD17	1232
□	CDH_SS_FMUSOHWD18	0

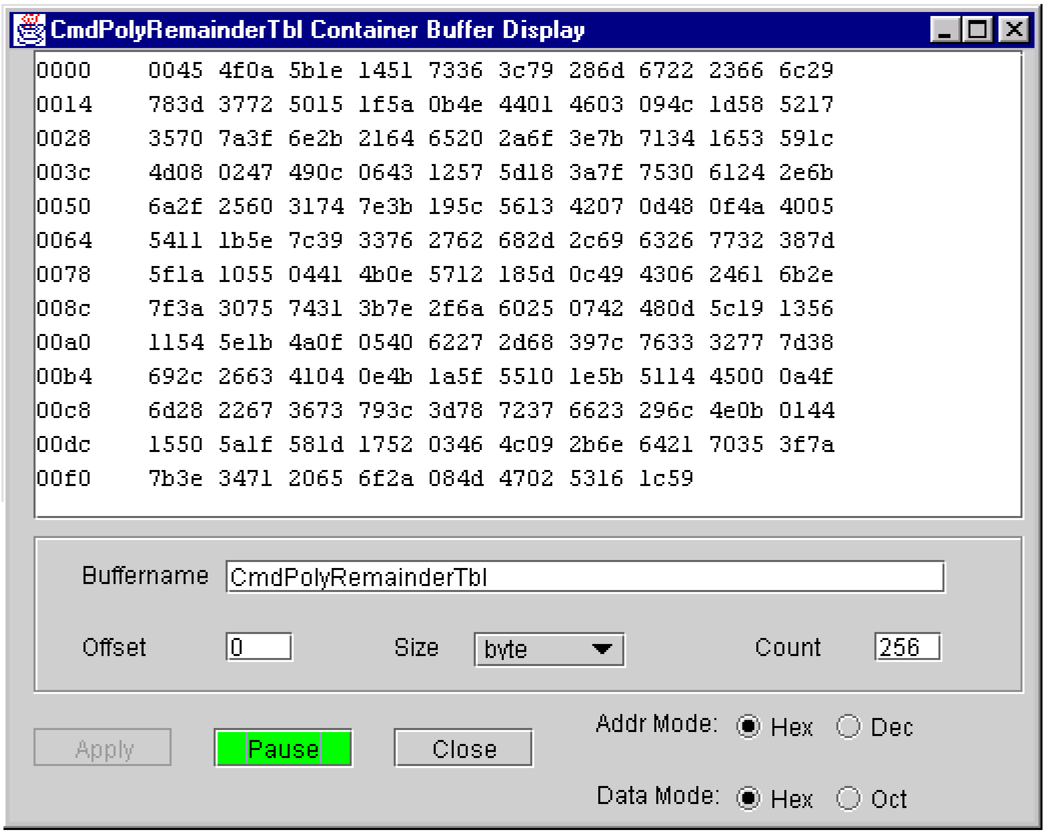
This display allows the user to view and modify individual items from the SCAURA module's container. Since telemetry points are stored as container items, telemetry points may be monitored and overwritten with this display. To request an item for display or editing, enter its name in the mnemonic field and click the **Apply** button. If there are any problems with displaying or modifying items, error messages are sent to the event message area of the screen. Multiple copies of this display are allowed, but each container item should only appear on one screen for properly updated values. The list of displayed container item names may be saved to a file using the **Save** button. Similarly, a saved set of item names may be loaded into this display by using the **Restore** button and selecting a saved file.

Field	**Description**
Checkbox	Click on this box to modify an item’s value.
Mnemonic	This field identifies a container item for display. Refer to section “SCAURA-4.0 Container Items” for the container names of specific items. This field is not case-sensitive.
Value	If the associated checkbox is not checked, this is the current decimal value of the container item. If the checkbox is checked, this is a data entry field for the value of the item. Values may be entered in decimal, octal, or Hexadecimal. Negative values for Signed Integer or Floating Point container items may be entered by preceding the number with a minus sign. Octal and Hexadecimal entries will be converted to decimal.

SCAURA-5.2.1.1.7 Display Container Buffer

This display may be used to display container buffers that do not have a customized display available for them. Enter the buffer name, offset, size and count and click on the **Apply** button. If there are problems displaying a buffer, error messages are sent to the event message area of the screen.

If the buffer's contents are updated frequently, the **Pause** button may be used to display a "snapshot" of the buffer. (Buffer contents will continue to be updated offscreen.) The button's name is changed to **Cont**. To resume updates of the buffer, press the **Cont** button. Only one copy of this display may be active at a time. When switching from the display of one buffer to another, it may be helpful to toggle the **Pause/Cont** button on and off to force a refresh of the buffer contents.



```
0000     0045 4f0a 5b1e 1451 7336 3c79 286d 6722 2366 6c29
0014     783d 3772 5015 1f5a 0b4e 4401 4603 094c 1d58 5217
0028     3570 7a3f 6e2b 2164 6520 2a6f 3e7b 7134 1653 591c
003c     4d08 0247 490c 0643 1257 5d18 3a7f 7530 6124 2e6b
0050     6a2f 2560 3174 7e3b 195c 5613 4207 0d48 0f4a 4005
0064     5411 1b5e 7c39 3376 2762 682d 2c69 6326 7732 387d
0078     5f1a 1055 0441 4b0e 5712 185d 0c49 4306 2461 6b2e
008c     7f3a 3075 7431 3b7e 2f6a 6025 0742 480d 5c19 1356
00a0     1154 5e1b 4a0f 0540 6227 2d68 397c 7633 3277 7d38
00b4     692c 2663 4104 0e4b 1a5f 5510 1e5b 5114 4500 0a4f
00c8     6d28 2267 3673 793c 3d78 7237 6623 296c 4e0b 0144
00dc     1550 5a1f 581d 1752 0346 4c09 2b6e 6421 7035 3f7a
00f0     7b3e 3471 2065 6f2a 084d 4702 5316 1c59
```

Buffername: CmdPolyRemainderTbl

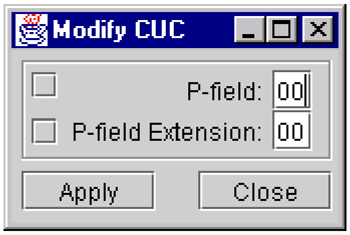
Offset: 0 Size: byte Count: 256

Addr Mode: ● Hex ○ Dec

Data Mode: ● Hex ○ Oct

Field	Description
Buffername	This field identifies the container buffer for display. Refer to section “SCAURA-4.0 Container Items” for names of container buffers.
Offset	This field specifies the first byte to be displayed. This display is limited to 1400 data bytes at a time. This field is used with the count field to specify the portion of the buffer to display.
Size	Size of the data items to display (choices are byte, word, double)
Count	Number of data items to display (limited to 1400 total characters)
Addr Mode	Addresses of the data may be displayed in hexadecimal or decimal
Data Mode	Data may be displayed in hexadecimal or octal formats

SCAURA-5.2.1.1.8 Modify CCSDS Unsegmented Time Code



□ P-field: 00

□ P-field Extension: 00

This display allows the user to modify the fixed fields of the CCSDS Unsegmented Time Code (CUC) in the telemetry packet headers. Clicking the enable boxes at left indicate whether or not the given field should be modified. Clicking the **Apply** button puts the settings into effect for all APIDs.

Data Entry Field	Description
P-field	P-field flags in the CUC. Enter hexadecimal value.

P-field Extension	Second byte of p-field, containing number of leap seconds since 1958. Enter hexadecimal value.

SCAURA-5.2.1.1.9 APID Status Display

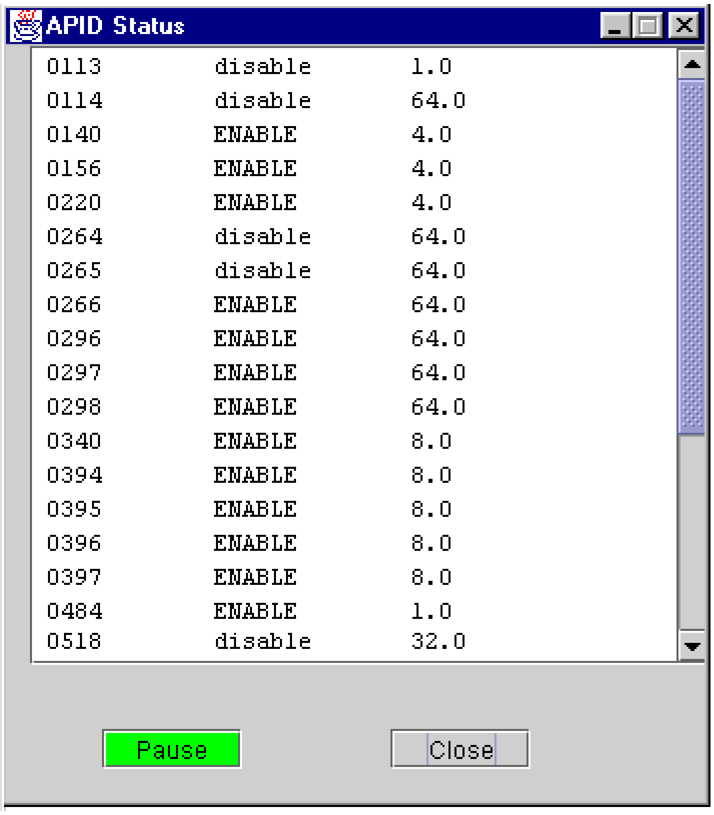
This dynamically updating display provides a scrollable list of the current status of all telemetry packets created from the project database. The first column provides the APID number in decimal. Next the word "disabled" or "ENABLED" is displayed. The last column provides the interval between packet transmissions in seconds.

The display's updating may be interrupted by clicking on the **Pause** button. The button's title will change to **Cont**. Click on the **Cont** button to resume status updates. Use the **Close** button to dismiss the display.

The "Control Packet…" telemetry menu item may used to modify any packet's status. Directives or scenario files may also be used to change packet status. The container names for these fields are TlmPacket<APID>Enabled and TlmPacket<APID>Interval, where the <APID> portion is replaced with the packet's APID in 4 decimal digits. For example, the following directives change the interval of packet 220 to 8 seconds and enable its transmission.

set TlmPacket0220Interval 8.0

set TlmPacket0220Enabled 1



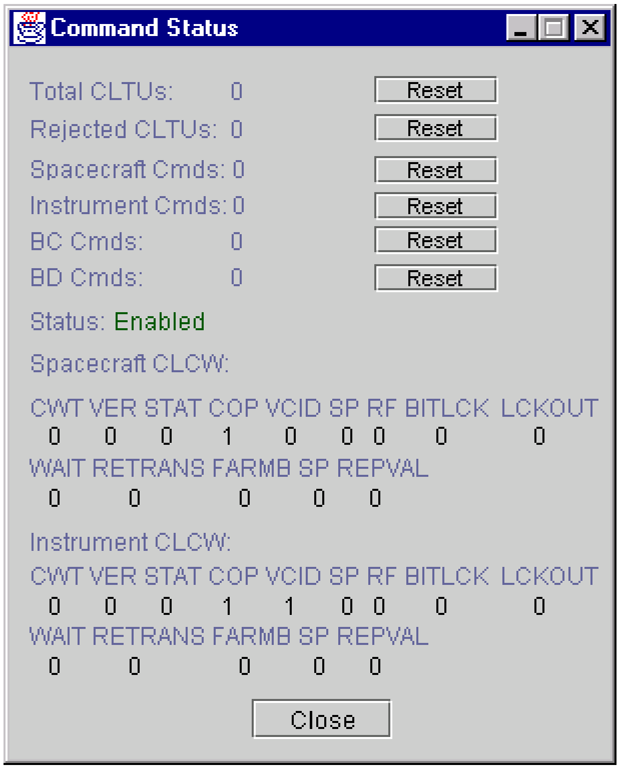
```
0113          disable       1.0
0114          disable       64.0
0140          ENABLE        4.0
0156          ENABLE        4.0
0220          ENABLE        4.0
0264          disable       64.0
0265          disable       64.0
0266          ENABLE        64.0
0296          ENABLE        64.0
0297          ENABLE        64.0
0298          ENABLE        64.0
0340          ENABLE        8.0
0394          ENABLE        8.0
0395          ENABLE        8.0
0396          ENABLE        8.0
0397          ENABLE        8.0
0484          ENABLE        1.0
0518          disable       32.0
```

SCAURA-5.2.1.2 Command Menu

Clicking the **Command** button on the main display brings up the command menu.

Menu item	Description
Start	Enable command reception
Stop	Disable command reception
Display Status…	Enable/disable status, command counts, CLCWs by virtual channel
Modify Validation Criteria…	Allows modification of validation options
Display Spacecraft Packet…	Displays spacecraft virtual channel packet contents
Display Instrument Packet…	Displays instrument virtual channel packet contents
Override CLCWs	Edit the fields of the Command Link Control Words
Display Spacecraft Load…	Display the Spacecraft Memory Load Buffer

SCAURA-5.2.1.2.1 Display Command Status



Total CLTUs: 0
Rejected CLTUs: 0
Spacecraft Cmds: 0
Instrument Cmds: 0
BC Cmds: 0
BD Cmds: 0
Status: Enabled

Spacecraft CLCW:

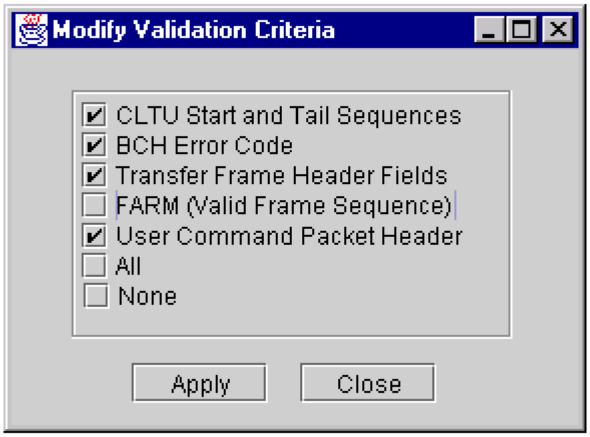
```
CWT VER STAT COP VCID SP RF BITLCK  LCKOUT
 0    0    0    1    0   0  0   0        0
WAIT RETRANS FARMB SP REPVAL
 0     0       0    0   0
```

Instrument CLCW:

```
CWT VER STAT COP VCID SP RF BITLCK  LCKOUT
 0    0    0    1    1   0  0   0        0
WAIT RETRANS FARMB SP REPVAL
 0     0       0    0   0
```

When the "Display Status…" option is selected, this display is shown. The command status display shows command counters, command enabled or disabled status and the Command Link Control Words (CLCWs) for the spacecraft and instrument virtual channels. Clicking the **Reset** button sets the counter in that row to zero.

SCAURA-5.2.1.2.2 Modify Validation Criteria



☑ CLTU Start and Tail Sequences
☑ BCH Error Code
☑ Transfer Frame Header Fields
☐ FARM (Valid Frame Sequence)
☑ User Command Packet Header
☐ All
☐ None

When the “Modify Validation Criteria” option is selected, this screen is displayed. The operator may select any combination of the validation options. The operator may enable all of the validation tests by selecting the “All” option or disable all of the validation tests by selecting the “None” option. The **Apply** button is used to activate the new settings. The **Close** button is used to dismiss the screen without changes.

SCAURA-5.2.1.2.2.1 None Validation Option

Even when the “None” option is selected the following validation tests are performed.

- A received CLTU must be long enough to contain a start sequence, a single codeblock, and a tail sequence or it will be rejected and an error event message will be generated.
- A CLTU’s length will be checked to see there are enough bytes for a start sequence, tail sequence, and an even multiple of codeblocks. If there are extra bytes, they will be reported in a warning event message. The CLTU will then be processed as if there were no extra bytes. The existence of “extra” bytes may indicate a problem with the source system’s formatting of the CLTU.
- The transfer frame header length field must report a value big enough to contain a transfer frame header and at least one byte of data or it will be rejected and an error event message will be generated.
- The transfer frame header length field is compared to the actual length of the passed frame buffer. If there are more data bytes than are reported in the header, these bytes are compared to the fill data byte. If there are any “extra” bytes that are not fill data, a warning event message will be generated. The transfer frame will then be processed and the “extra” bytes will be ignored. The existence of “extra” non-fill bytes may indicate a problem with the source system’s formatting of the Transfer Frame.
- The transfer frame header VCID is used to determine whether the frame should be processed as a critical TIE command or passed to the spacecraft or instrument virtual channel Frame Acceptance and Reporting Mechanism (FARM). The VCID must match one of the four expected VCID values stored in the container. When a VCID does not match any of the expected values, the transfer frame is rejected and an error event message is generated.

SCAURA-5.2.1.2.2.2 Validation of CLTU Start and Tail Sequences

Since a received CLTU must contain a valid start and tail sequence to be recognized by the software, disabling validation of these items is not an option. Rather, the software will report appropriate event error messages if the start sequence cannot be located, or if the start sequence is located but the tail sequence is not found.

SCAURA-5.2.1.2.2.3 BCH Error Code Validation Option

When Bose-Chaudhuri-Hocquenghem (BCH) Error Code validation is enabled, the parity byte of each received codeblock is compared to the parity value calculated from the codeblock data area. If a parity comparison fails, both parity bytes are reported in an error event message and the current CLTU is rejected.

SCAURA-5.2.1.2.2.4 Transfer Frame Header Validation Option

When transfer frame validation is enabled, the following validations are done.

- The transfer frame header version field must contain 0 or the frame is rejected and an error event message is generated.
- The transfer frame header spare field must contain 0 or the frame is rejected and an error event message is generated.
- The transfer frame header length is tested to be within the range specified in the ICD. If the length is invalid, the “Incorrect Frame Length” status is stored in the CLCW, the frame is rejected, and an error event message is generated.
- The transfer frame header length is compared to the actual length of the received buffer. If the frame contains fewer bytes than reported in the header, the “Incomplete Frame” status is stored in the CLCW, the frame is rejected, and an error event message is generated.
- The transfer frame header mode flags are checked. If the mode is “AC”, the “Illegal frame type (Type-AC)” status is stored in the CLCW, the frame is rejected, and an error event message is generated.
- If the transfer frame header mode flags specify “BC”, the frame data is checked for a valid transfer frame control command. If the frame data does not contain a valid frame control command, the “Illegal Type-BC frame” status is stored in the CLCW, the frame is rejected, and an error event message is generated.
- The transfer frame header VCID field is compared to the VCID field from the FARM’s CLCW. If the VCID fields do not match, the “Illegal VCID” status is stored in the CLCW, the frame is rejected, and an error event message is generated.
- If the transfer frame header SCID field does not match the Aura SCID stored in container point CmdSCID, the “Illegal SCID” status is stored in the CLCW, the frame is rejected, and an error event message is generated.

SCAURA-5.2.1.2.2.5 Farm (Valid Frame Sequence) Validation Option

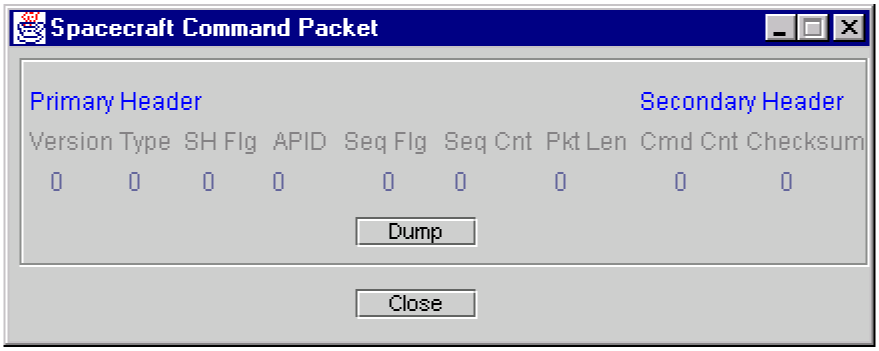
When FARM validation is enabled, the transfer frame header sequence field is subjected to the FARM-1 protocol and the CLCW flags, FARM-B Counter, and Report Value fields will be updated accordingly. Event messages are generated for every transfer frame that fails the acceptance test.

SCAURA-5.2.1.2.2.6 User Command Packet Header Validation Option

When packet validation is enabled, fields in the packet header will be validated as specified in the EOS Aura Spacecraft to Ground ICD.

SCAURA-5.2.1.2.3 Display Spacecraft Packet

This display shows the spacecraft virtual channel command packet header. A dump of the full packet buffer may be requested by pressing the **Dump** button.



Spacecraft Command Packet

Primary Header							Secondary Header	
Version	Type	SH Flg	APID	Seq Flg	Seq Cnt	Pkt Len	Cmd Cnt	Checksum
0	0	0	0	0	0	0	0	0

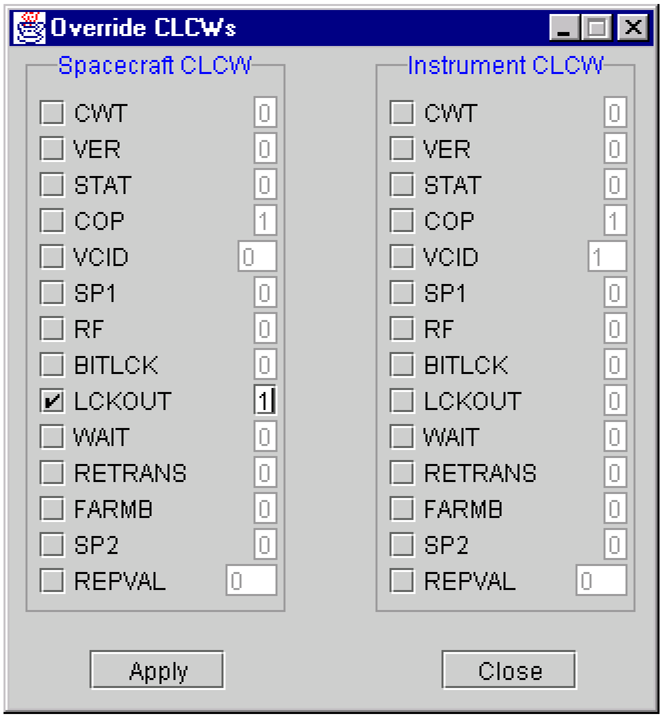
The Dump address region may be displayed in decimal or hexadecimal. The data region may be displayed in hexadecimal or octal. Since this display normally updates as the buffer contents are changed, use the **Pause** button to freeze the current contents. Use of the **Pause** button does not affect command packet reception. When the **Pause** button has been used, the button's label is changed to **Cont** (for continue). Press the **Cont** button to resume screen updates. Note that the display will be updated to show the most recent command packet received. Intermediate packets received while the display was frozen can not be displayed.

SCAURA-5.2.1.2.4 Display Instrument Packet

This display shows the instrument virtual channel packet header. Optionally, a dump of the full packet buffer may be requested. They are very similar in appearance and function to the spacecraft packet header and packet dump displays that are described in the previous section.

SCAURA-5.2.1.2.5 Override CLCWs

This display shows the Command Link Control Words (CLCWs) for the two virtual channels broken out into bit fields. Any field may be overridden by checking the field's selection box and then typing in the new decimal value. The new value takes affect when the **Apply** button is pressed. Note that since these are bit fields, attempts to assign values that are too large will result in truncation of the value. Review changes on the CLCW portion of the Command Status display. Event messages also show the values assigned.

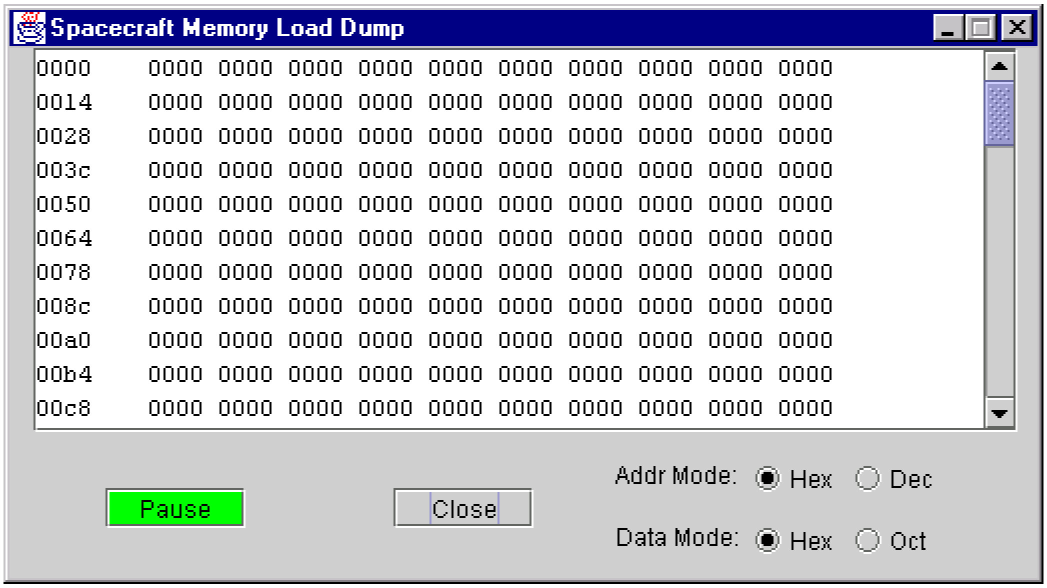


Spacecraft CLCW		Instrument CLCW	
□ CWT	0	□ CWT	0
□ VER	0	□ VER	0
□ STAT	0	□ STAT	0
□ COP	1	□ COP	1
□ VCID	0	□ VCID	1
□ SP1	0	□ SP1	0
□ RF	0	□ RF	0
□ BITLCK	0	□ BITLCK	0
☑ LCKOUT	1	□ LCKOUT	0
□ WAIT	0	□ WAIT	0
□ RETRANS	0	□ RETRANS	0
□ FARMB	0	□ FARMB	0
□ SP2	0	□ SP2	0
□ REPVAL	0	□ REPVAL	0

CLCW Field	Description
CWT	Control Word Type (1 bit)
VER	Version (2 bits)
STAT	Status (3 bits)
COP	Command Operations Procedure (COP) in Effect (2 bits)
VCID	Virtual Channel Identification (6 bits)
SP1	Spare field 1 (2 bits)
RF	No RF Available Flag (1 bit)
BITLCK	No Bit Lock (1 bit)
LCKOUT	Lockout Flag (1 bit)
WAIT	Wait Flag (1 bit)
RETRANS	Retransmission Flag (1 bit)
FARMB	FARM-B Counter (2 bits)
SP2	Spare field 2 (1 bit)
REPVAL	Report Value (8 bits)

SCAURA-5.2.1.2.6 Display Spacecraft Load

When the “Display Spacecraft Load…” option is selected, the buffer containing the accumulated spacecraft load packet data is displayed.



Spacecraft Memory Load Dump

```
0000     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0014     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0028     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
003c     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0050     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0064     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0078     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
008c     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
00a0     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
00b4     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
00c8     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
```

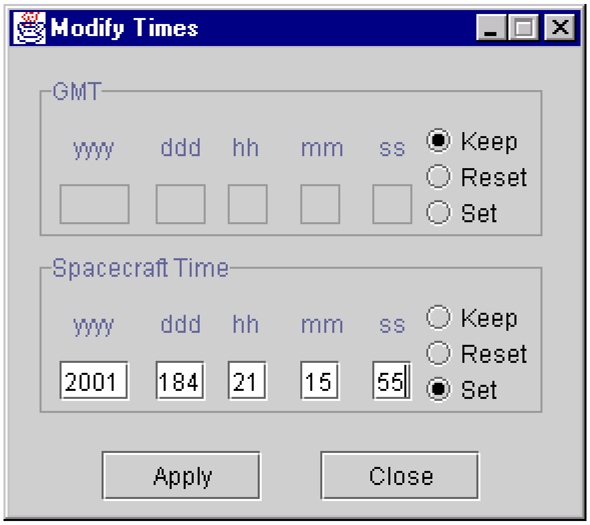
Addr Mode: ● Hex ○ Dec

Data Mode: ● Hex ○ Oct

SCAURA-5.2.1.3 Time

The time area of the main display shows the current Greenwich Mean Time (GMT) and spacecraft time fields. The GMT is a reflection of the simulated time of the SCAURA module, i.e., what time will be used to model external events (transitions to day or night, for example) within the module. It is not currently used. The spacecraft time is the time that the spacecraft thinks it is. In particular, it is the time (in the appropriate format) that goes into the secondary header of the telemetry packets generated by the SCAURA module. Clicking the **Time** button brings up a menu with one selectable item, "Modify Times…".

SCAURA-5.2.1.3.1 Modify Times Display



Modify Times

GMT

yyyy	ddd	hh	mm	ss	
					● Keep ○ Reset ○ Set

Spacecraft Time

yyyy	ddd	hh	mm	ss	
2001	184	21	15	55	○ Keep ○ Reset ● Set

This display allows the user to set the GMT or spacecraft time manually, retain the time currently being used, or reset the time to the current system time.

Clicking the **Apply** button will cause the requested actions to take effect.

Clicking the **Close** button closes the window with no further action taken.

Field	Description
yyyy	Year
ddd	Day of year
hh	Hour of day
mm	Minute of hour
ss	Second of minute
Keep	Don’t update time
Reset	Change time to current system time
Set	Set time to value supplied

SCAURA-5.3 Remove

Clicking the “Remove” option from the module pop-up menu causes this module to be removed from the project. This option is not available during run-time.

SCAURA-5.4 About

Clicking the “About” option from the module pop-up menu requests a display that lists generic information about the module such as the numbers of input and output channels.

SCAURA-6.0 Special Operating Instructions

SCAURA-6.1 Serial Mode Operation

In order to run the SCAURA module in serial mode select serial mode from the Select Simulation Mode Display during project configuration. SCAURA output channel 3 must be linked to a serial output module. A serial I/O card is required on the server platform.

SCAURA-6.2 Triggering Scenarios with Commands

A powerful feature of the SCAURA module is to be able to automatically trigger the execution of specified scenario files when specific commands are recognized from the database. The project must contain a scenario module. The output channel from the scenario module must be connected to input channel 1 of the SCAURA module.

SCAURA-6.2.1 Using the Default Scenario Directory

Create a file named **Command-Scenario.txt** in the release's scenario directory. (The scenario directory should be at the same level as the properties, elog and save directories.) The scenario files to be executed should be placed in this directory. The command trigger definition file is read immediately after the database is loaded. The trigger information is stored in memory. Event messages will be generated for any invalid command mnemonics and any invalid scenario filenames in the definition file.

Format of Command-Scenario.txt File

Command-scenario pairs are specified by the following format:

```
      command mnemonic  |  scenario filename
```

One pair is specified per line with the "|" character as the delimiter. Leading and trailing spaces and tab characters will be removed so these characters may be used to align the data. The scenario filename may be specified as a full or relative pathname. Relative pathnames will start from the scenario directory. Each command mnemonic should only appear in the definition file once.

Any line starting with a semicolon ";" in column one is ignored as a comment.

Blank lines are also ignored.

SCAURA-6.2.2 Using an Alternate Scenario Directory

If a user wishes to use an alternate set of scenario files, place them in a specific directory on the disk, for example, D:\My-Scenarios. Place the command trigger definition file, Command-Scenario.txt, in this directory as well.

Every reference to a scenario file, from within the command trigger definition file or from within a scenario file MUST use the file's fully specified path name. For example, an entry in the definition file might be:

```
      CDH_EXECUTE_TIENOP | D:\My-Scenarios\tienop.txt
```

If any scenario file calls tienop.txt, the format for the calling statement within the scenario file is

```
      Start scenario D:\My-Scenarios\tienop.txt
```

When the simulator is started, the default command trigger definition file will be read from the default directory. To force replacement of the default trigger information with the alternate definition file send directives similar to the following to the SCAURA module. Do not use quotes around the alternate directory name. The ending slash is required syntax.

```
      Set scenarioPath D:\My-Scenarios\

      Set readScenFile 0
```

These set directives must be done each time the simulator is started. To make this easier, place both directives in a scenario file in the alternate directory. After the simulator is running, use the Scenario module GUI file browser to navigate to the alternate directory and execute this file.

SCAURA-6.2.3 Update/Replace Command Trigger Definitions

During run-time the operator may request that the Command-Scenario.txt definition file be read in again replacing the previous definitions. On the directive line to the SCAURA module, enter the following. Check the event message log for status and error messages.

```
      Set ReadScenFile 0
```

Internet Protocol (IP) Modules

IP-1.0 Overview

The Input and Output Internet Protocol (IP) modules receive/send data packets from/to other sources/destinations using one of several IP types (TCP/IP-Client, TCP/IP-Server, UDP Multicast, or UDP Unicast). The Input IP module receives data from an external source and passes that data to another module. The Output IP module receives data from a module and can pass that data to another module and send it to an external destination. Both the Input and Output IP modules are discussed in this one section because of their great similarity.

IP-2.0 Inputs

The Input IP module does not have any input channels. The Output IP module has a single input channel, which is listed below.

Channel	Data expected	Validation performed	Processing performed
1	Packets	None	Received data is passed to connected modules and transmitted to configured external IP destination

IP-3.0 Outputs

Both the Input IP and Output IP modules have a single output channel.

Channel	Description
1	Input IP module passes received data to all connected modules. Output IP module passes received data to connected modules (usually the Log Module) and sends it to an external destination

IP-4.0 Container Items

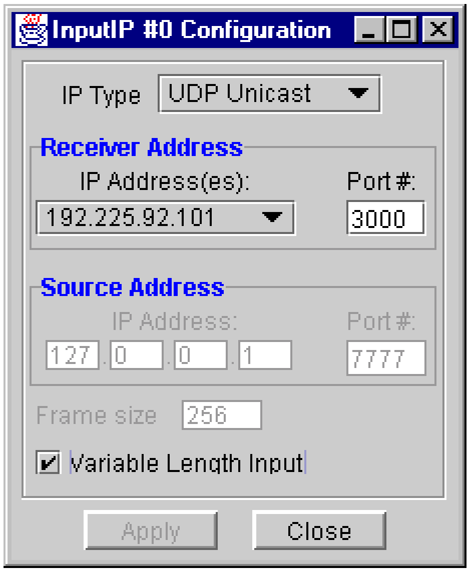
IP module container items are not accessible via operator directives, so they are not listed here.

IP-5.0 Displays

Click in the center of the module in the project window to access displays. The following pop-up menu will appear. The “Remove” option can be used during project design to remove this module. The “Configure” option must be used prior to running the project and is unavailable at run-time.

Module Pop-Up Menu Item	Description
Configure	Access the configuration display
Run-time	Access the Run-time menu for the module
Remove	Remove module from the project
About	Display generic module information

IP-5.1 Configuration Menu



InputIP #0 Configuration

IP Type: UDP Unicast

Receiver Address

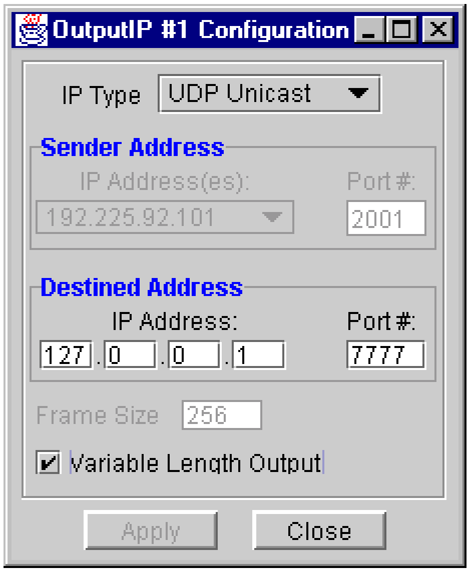
IP Address(es): 192.225.92.101 Port #: 3000

Source Address

IP Address: 127 . 0 . 0 . 1 Port #: 7777

Frame size 256

☑ Variable Length Input



OutputIP #1 Configuration

IP Type: UDP Unicast

Sender Address

IP Address(es): 192.225.92.101 Port #: 2001

Destined Address

IP Address: 127 . 0 . 0 . 1 Port #: 7777

Frame Size 256

☑ Variable Length Output

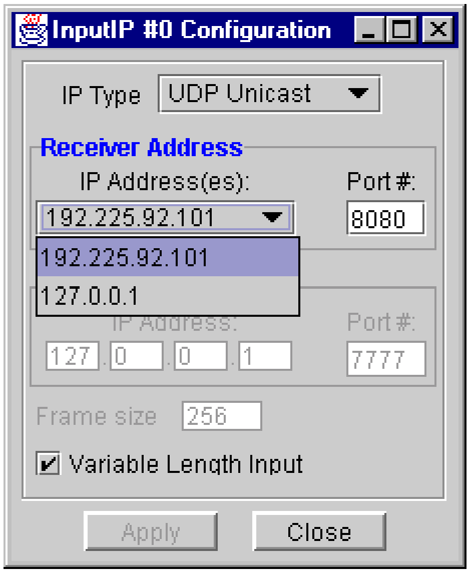
Clicking on the “Configure” item of the Input (or Output) IP module pop-up menu produces a display similar those shown above.

IP-5.1.1 IP Type Field

There are 4 possible IP Types available from a drop down menu: UDP Multicast, TCP/IP-Client, TCP/IP-Server, and UDP Unicast.

IP-5.1.2 Receiver (Sender) Address Box

Various options will be available depending on the configuration of the host PC running the server.



IP Type: UDP Unicast

IP Address(es): 192.225.92.101

Port #: 8080

192.225.92.101
127.0.0.1

IP Address: 127 . 0 . 0 . 1

Port #: 7777

Frame size: 256

☑ Variable Length Input

Step 1. If there is more than one Ethernet Card on the host machine where the server is running, the user can select an IP address from the drop down menu (similar to that shown on the left) as the network interface for the host machine.

Step 2. Certain IP Types will require a Port Number. If this option is sensitized, enter a Port Number for the host machine to bind to, to either receive or offer a service.

Note: The IP address **127.0.0.1** is a loopback address. It is used to test the network program in a single machine (except in UDP Multicast mode). **Don't** select **127.0.0.1** as a receiver (or sender) IP address in UDP Multicast mode.

IP-5.1.3 Source (Destination) Address Box

There are different settings for the different IP types available.

For use in **TCP/IP-Client** mode:

Step 1. Enter the four fields of the Server IP Address to which the client will attempt to connect.

Step 2. Enter the Port Number where the server is offering the service.

In **TCP/IP-Server** mode the user is not allowed to configure this box. The four fields of IP address and 1 field of Port Number are desensitized.

For use in **UDP Unicast** mode (Output IP module only):

Step 1. Enter the four fields of the destination IP address of the remote host where the packets will be sent.

Step 2. Enter the Port Number where the packets will be sent.

The user is not allowed to configure this box when using an Input IP module in **UDP Unicast** mode. The fields will be desensitized.

.

For use in **UDP Multicast** mode (Input IP module):

Enter the four fields of the Multicast Group Address. This is the multicast group address that the Input IP module will join and receive packets from. The Port Number is not configurable. It will be desensitized. Enter the port number of the host PC in the Receiver Address box.

For use in **UDP Multicast** mode (Output IP module):

Step 1. Enter the four fields of the Multicast Group Address. This is the multicast group address where the packets will be sen.

Step 2. Enter the Port Number where the packets will be sent.

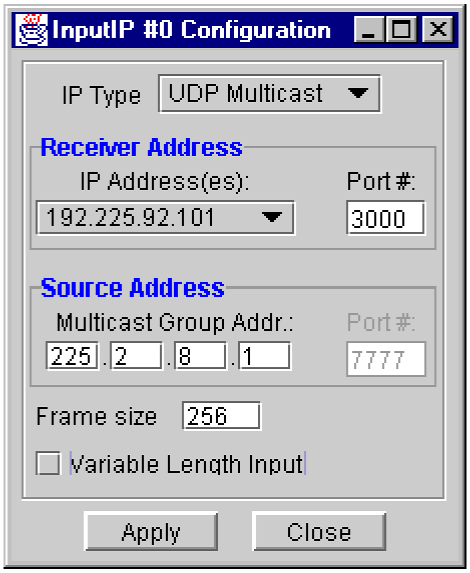
Note: The valid range of port numbers is 1000 to 65535. The valid range of multicast group IP addresses is 224.0.1.0 through 239.255.255.255.

IP-5.1.6 Variable Length Input (Output) Check Box

When this option is selected, the box will be checked and the Frame Size field will be desensitized. The actual length of each data packet will be used without truncation or padding. This is the default configuration.

IP-5.1.7 Frame Size Field

When the Variable Length Input (Output) Check Box is deselected, the Frame Size field becomes active. The user should enter the desired number of bytes to receive (transmit). The default is 256 bytes.



InputIP #0 Configuration

IP Type: UDP Multicast

Receiver Address

IP Address(es): 192.225.92.101

Port #: 3000

Source Address

Multicast Group Addr.: 225 . 2 . 8 . 1

Port #: 7777

Frame size: 256

□ Variable Length Input

If the size selected is larger than the actual data, the additional bytes will be filled with binary zero.

If the Frame Size selected is smaller than the actual data size, the data bytes received (transmitted) will be truncated and the additional bytes will be lost.

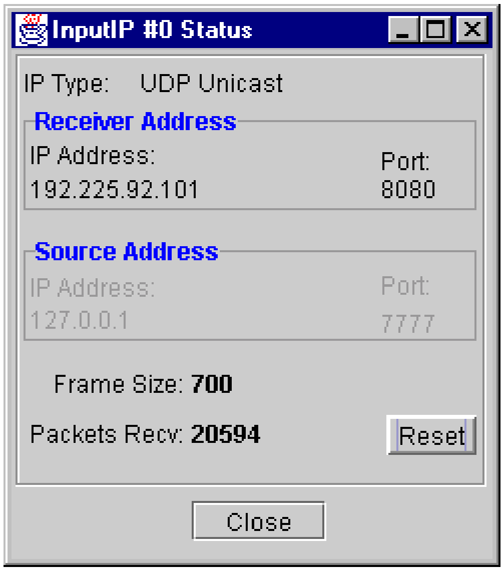
IP-5.2 Run-time Menu

The Run-time menu for the IP modules contains the following four items.

Run-time Menu Item	Description
Show Status	Request Status Display
Show Raw Packet	Request Raw Packet Display
Pause (or Resume)	Pause (or Resume) the module
Stop (or Restart)	Stop (or Restart) the module

IP-5.2.1 Show Status

Select the “Show Status” item to request the module’s status. This produces a display similar to one shown below.



InputIP #0 Status

IP Type: UDP Unicast

Receiver Address

IP Address: 192.225.92.101 Port: 8080

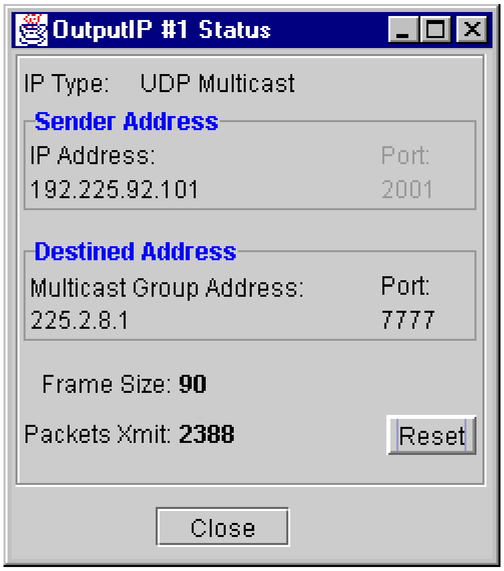
Source Address

IP Address: 127.0.0.1 Port: 7777

Frame Size: **700**

Packets Recv: **20594**

OR



OutputIP #1 Status

IP Type: UDP Multicast

Sender Address

IP Address: 192.225.92.101 Port: 2001

Destined Address

Multicast Group Address: 225.2.8.1 Port: 7777

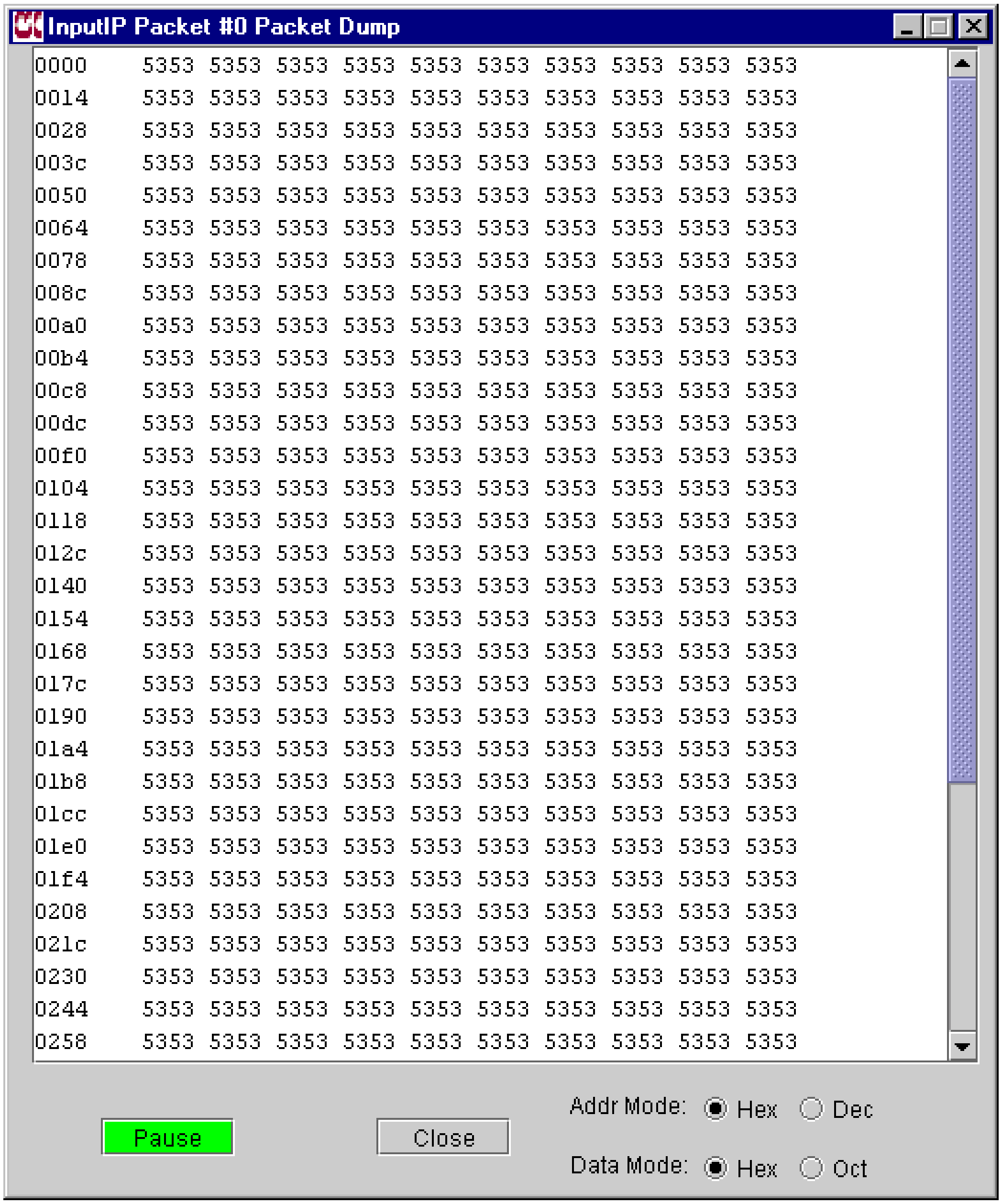
Frame Size: **90**

Packets Xmit: **2388**

This display shows the Input (or Output) IP module’s IP Type, Receiver (Sender) Address (including the port number, if appropriate for the mode), Source (Destination) Address (IP address, or multicast group IP address and port number), Frame Size, and number of packets received (or transmitted). If variable length items are being received, the Frame Size field will show the size of the most recent packet received (transmitted). The **Reset** button will reset the count of packets received. Click the **Close** button to dismiss the display.

IP-5.2.2 Show Raw Packet

Select “Show Raw Packet” from the Run-time Menu to view the most recent data packet received by the module. An example of the packet display follows.



InputIP Packet #0 Packet Dump

```
0000     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
0014     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
0028     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
003c     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
0050     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
0064     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
0078     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
008c     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
00a0     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
00b4     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
00c8     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
00dc     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
00f0     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
0104     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
0118     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
012c     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
0140     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
0154     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
0168     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
017c     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
0190     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
01a4     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
01b8     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
01cc     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
01e0     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
01f4     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
0208     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
021c     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
0230     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
0244     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
0258     5353 5353 5353 5353 5353 5353 5353 5353 5353 5353
```

Addr Mode: ● Hex ○ Dec

Data Mode: ● Hex ○ Oct

The address region may be displayed in decimal or hexadecimal by clicking on the respective radio button. Likewise, the data region may be displayed in hexadecimal or octal. Since this display updates as the buffer contents are changed; the **Pause** button may be used to freeze the current contents. Use of the **Pause** button does not affect data transmission or reception. When the **Pause** button has been used, its label is changed to **Cont** for continue. Press the **Cont** button to resume screen updates. Note that the display will update with the most recent packet processed. Packets processed by the module while the display was frozen can not be displayed. Click the **Close** button to dismiss the display.

IP-5.2.3 Pause

Select “Pause” from the Run-time Menu to pause the IP module’s processing. The color around the module’s border will change from green-striped (indicates run state) to red-striped (indicates non-run state). This option's text will be changed to “Resume”. When an Input IP module is paused, a limited amount of data received will be queued up and sent on when the module is Resumed. When an Output IP module is paused, all data passed to it is dropped.

IP-5.2.4 Resume

After an individual module has been paused, select “Resume” from the Run-time Menu to resume the module's execution. The color around the module’s border will change from red to green. This option's text will be changed to “Pause”. The Input IP module first processes any data queued up while paused before forwarding new inputs. The Output IP module processes new data only.

IP-5.2.5 Stop

Select “Stop” from the Run-time Menu to stop the IP module’s processing. The color around the module’s border will change from green to red. This option's text will be changed to “Restart”. Once the module has been stopped, the “Configure” option of the module pop-up menu is available again and the module may be reconfigured. Any data passed to stopped modules is discarded.

IP-5.2.6 Restart

After an individual module has been stopped, select “Restart” from the Run-time Menu to restart the module. The color around the module’s border will change from red to green. This option's text will be changed back to “Stop”. Since their configurations may have changed, the IP modules reset all counters and only process new data.

IP-5.3 About

Selecting the “About” option from an IP module pop-up menu produces a display that lists the module’s number of inputs and outputs, whether directives are allowed, names of authors and the version number.

IP-6.0 Special Operating Instructions

There are no special operating instructions for this release.

Log Module

Log-1.0 Overview

The Log module writes the data it receives to a specified log file.

Viewing or printing of log files may be accomplished by means of an external program. The software used must be capable of displaying binary data in an ASCII representation. The shareware utility, Hexedit, which is available over the Internet from Alexander Reidel Informations-Systeme, is one such program.

Log-2.0 Inputs

Ch	Data expected	Validation performed	Processing performed
1	bytes	None	Received log data is written into the log file.

Log-3.0 Outputs

The Log module does not have any output channels.

Log-4.0 Container Items

Name	Type	Read-Only	Description
LOGFILENAME	SimString	No	Logging file name
LOGFIXBOOL	UnsByte	No	1 for fixed length logging, 0 for variable length logging
LOGPSIZE	UnsWord	No	Packet size for fixed length logging only
LOGHEADERBOOL	UnsByte	No	1 for logging with header, 0 for logging without header
LOGMSIZE	UnsDword	No	Max. number of bytes that the log file can be written into
LOGBLOGGED	UnsDword	No	Number of bytes that are written into the log file
LOGNPACKET	UnsDword	No	Number of packets that are written into the log file

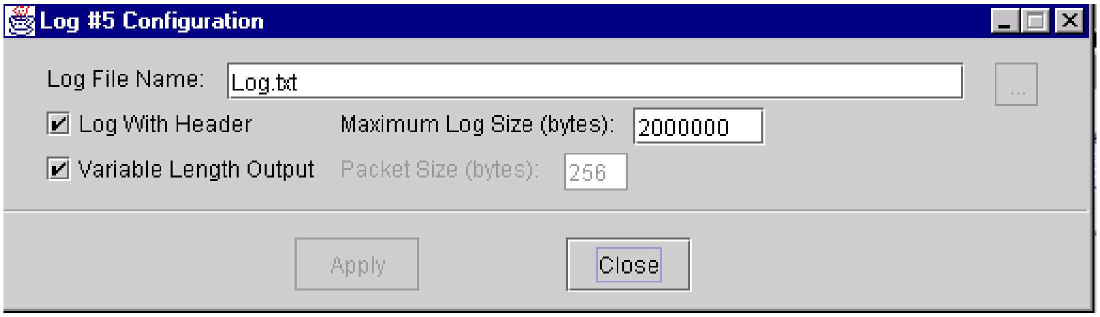
Log-5.0 Displays

To access displays for this module, click in the center of the Log module in the project window. The following pop-up menu choices will appear. The “Remove” option can be used during project design to remove this module. The “Configure” option should be used prior to running the project and is unavailable at run-time. The Run-time option is available only when the project is running.

Module Pop-Up Menu Item	Description
Configure	Access the configuration display
Run-time	Access the Run-time menu for the module
Remove	Remove the module from the project
About	Display Information about the module

Log-5.1 Configuration

Selecting the “Configure” pop-up menu option produces a display similar to the following screen.



Log #5 Configuration

Log File Name: Log.txt

☑ Log With Header

Maximum Log Size (bytes): 2000000

☑ Variable Length Output

Packet Size (bytes): 256

Log-5.1.1 Log File Name

The Log File Name specifies where the log data is written. If this file already exists, its data will be overwritten. If there is more than one Log module in a project, they cannot write to the same log file.

Log-5.1.2 Maximum Log Size (bytes)

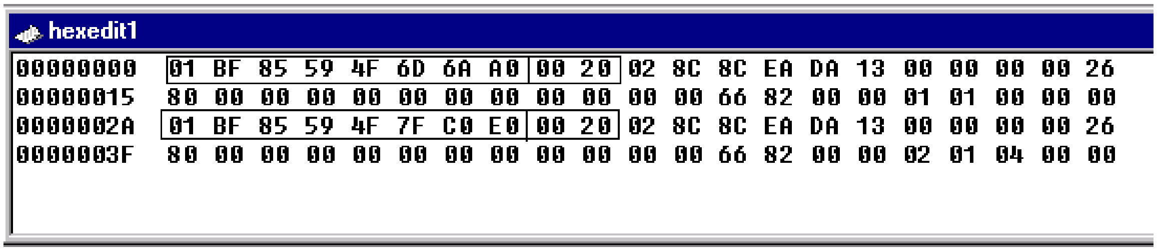
The Maximum Log Size in bytes limits the amount of storage used for logging. When this maximum is reached, the log file is closed and an event message is generated.

Log-5.1.3 Packet Size (bytes)

The Packet Size defines the size of the data buffer to be written to the log file. A packet size that is smaller than the actual data buffer received will result in truncation of the data that is logged. A packet size that is larger than the actual data buffer received will be zero filled to the packet size and then written to the log file.

Log-5.1.4 Log With Header

When the Log With Header box has been checked, all log records will begin with a log header of ten bytes. This header consists of eight bytes of system time followed by two bytes of data length. The following example shows two CLCW packets. The header in each packet has been circled for clarity. When the Log With Header box is unchecked, all records will be written without a header.



```
00000000  01 BF 85 59 4F 6D 6A A0 00 20 02 8C 8C EA DA 13 00 00 00 00 26
00000015  80 00 00 00 00 00 00 00 00 00 00 00 66 82 00 00 01 01 00 00 00
0000002A  01 BF 85 59 4F 7F C0 E0 00 20 02 8C 8C EA DA 13 00 00 00 00 26
0000003F  80 00 00 00 00 00 00 00 00 00 00 00 66 82 00 00 02 01 04 00 00
```

Log-5.1.5 Variable Length Output

When the Variable Length Output box has been checked, varying length records are written to the log file. No truncation or padding of received data is done. When this field is checked, the packet size field is desensitized.

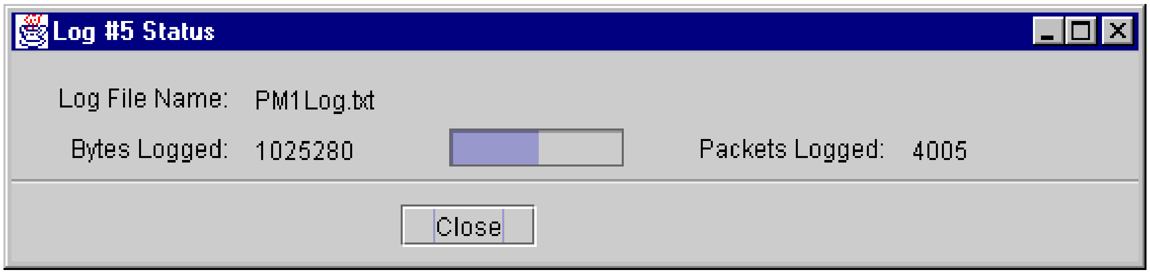
Log-5.2 Run-time

Clicking the “Run-time” option of the module pop-up menu produces a Run-time menu.

Run-time Menu Item	Description
Show Status	Request Status Display
Pause (or Resume)	Pause (or Resume) the module
Stop (or Restart)	Stop (or Restart) the module

Log-5.2.1 Show Status

When the “Show Status” option is selected from the Run-time Menu, a screen like the following is displayed. The bar in the center of the screen provides and indication of the percentage of the log area in use. The blue portion of the bar represents the logged data.



Log File Name: PM1Log.txt

Bytes Logged: 1025280

Packets Logged: 4005

Log-5.2.2 Pause/Resume

Select “Pause” from the Run-time Menu to pause the Log module’s processing. The color around the module’s border will change from green (indicates run state) to red (indicates stopped state). The text of this menu option will change to “Resume”. While the Log Module is paused, it does not receive or log data.

After a Log module has been paused, select “Resume” from the Run-time Menu to restore processing. The color around the module’s border will change from red to green. The text of this menu option will change to “Pause” and the module pop-up menu will have the normal Run-Time selections.

Log-5.2.3 Stop/Restart

Select “Stop” from the Run-time Menu to stop the Log module’s processing. The color around the module’s border will change from green to red. The text of this menu option will change to “Restart". Once the module has been stopped, the “Configure” option of the module pop-up menu is available again and the module may be reconfigured.

After a Log module has been stopped, select “Restart” from the Run-time Menu to restart the module's execution. The color around the module’s border will change from red to green. The text of this option will change to “Stop" and the module pop-up menu will have the normal Run-Time selections.

Log-5.3 About

Selecting the “About” option from the module pop-up menu produces a display that lists the module’s number of input links, number of output links, whether directives are allowed, names of authors and the version number.

Log-6.0 Special Operating Instructions

There are no special operating instructions for this release

Scenario Module

Scenario-1.0 Overview

The Scenario module reads directives from specified scenario files line by line and passes them to down-linked modules for processing. The user interface allows the operator to control and monitor the execution of up to five scenarios at a time. The scenario module may also receive scenario file names from a down-linked module. In addition, any scenario file being processed may start the execution of other scenario files.

Scenario-2.0 Inputs

Scenario-2.1 Server/Properties/Property.txt

When the project containing a scenario module is started, the server reads a property.txt file in the server's properties directory to configure several project parameters. One of these parameters, ***ConcurrentScenario***, defines how scenario modules will execute nested scenario files and module triggered scenario files.

The default setting for *ConcurrentScenario* is one. When set to one, the scenario module executes nested and external module requested scenario files concurrently. Additional system resources are dynamically allocated and released as these scenarios start and stop execution.

If the setting for *ConcurrentScenario* is zero, both types of non-GUI controlled scenario files are collected in a queue and executed in first-in-first-out (FIFO) order. This method is less taxing on system resources but may not execute scenario files quickly enough in some cases. This mode should be used if there are system resource problems over time or if the execution of scenario files causes degradation of other processing.

Scenario-2.2 Module Triggered Scenarios

The Scenario module does not technically have any input channels. It is not possible to create an input link to this module. Its lone output channel is actually bi-directional, providing the down-linked module a way to pass names of scenario files to be executed. Processing specific to the down-linked module may trigger the passing of scenario file names back to the scenario module for immediate execution.

Scenario-2.3 Scenario File Input

Much of the input to this module is via scenario files, which are read during run-time. There are no naming conventions for scenario files. Scenario files are expected to be in the scenario directory, which is below the server executable directory, unless a full pathname is specified. Relative pathnames are assumed to start from the scenario directory.

Scenario-2.3.1 Scenario File Contents

The format of ASCII text scenario files is very simple. Each line is limited to 260 characters. Each line contains one of the following items. Tab characters and blanks may be used to indent scenario lines for readability. Although Scenario-module specific key words are shown below in uppercase, they are not case sensitive.

Line	**Description**
comment line	A semicolon in the first column defines a comment.
blank line	A line containing only blank or tab characters is treated as a comment line.
SLEEP milliseconds	The scenario module pauses execution of the file for the number of milliseconds specified. When *milliseconds* are specified as -1, the directive is a SLEEP forever. For a GUI controlled scenario, a sleep forever is equivalent to the pause button being used. For non-GUI controlled scenarios, the SLEEP forever instruction is ignored.
START SCENARIO name	The scenario module starts execution of a nested scenario file. This nested file is not available for operator intervention.
directive	Any line that is not a comment or processed by the scenario module is assumed to be a directive for the linked module. The receiving module performs any validation done on this line. The format for directives is the same as if the operator had entered it in the project directive entry line.
IF (expression)	Specifies the conditional expression that must be true in order for the directives following this line up to the next ELSE IF, ELSE or ENDIF statement to be executed.
ELSE IF (expression)	Specifies a conditional expression that must be true in order for the directives following this line up to the next ELSE, ELSE IF or ENDIF statement to be executed.
ELSE	Specifies the beginning of the directives to be executed when the preceding IF or ELSE IF expression is false.
ENDIF	Specifies the end of any optional ELSE clauses and the end of the current IF statement.
WHILE (expression)	Specifies the conditional expression that must be true in order for the directives following this line up to the ENDWHILE statement to be executed. One of the directives within the loop body should be a SLEEP directive.
ENDWHILE	Specifies the end of the current WHILE loop.

Scenario-2.3.1.1 Specifying Modules to Receive Scenario Directives

A scenario file may contain directives for more than one module. Any executable line within a scenario file may start with an optional module number field with the format:

#x <directive line text>

where x is a number from 1 to 4. A directive labeled with #2 is sent to the module(s) connected to output channel 2. Subsequent directives are also sent to channel 2 until the end of the scenario file or another channel is specified. When present, the module number must be followed by a blank character. At the beginning of each scenario's execution, the module number is initialized to 1. In the absence of any module numbers, directives are sent to the module(s) linked to the scenario module's output channel 1.

Scenario-2.3.1.2 Specifying Conditional Expressions

The conditional expression must be specified on a single line of up to 260 characters (including the return key). The directives to be conditionally executed must be entered on the same line as the keywords IF and WHILE.

The special characters used to describe conditional expressions for IF and WHILE statements are parentheses, binary math operators (+, -, *, /, %), binary logical operators (<, <=, ==, >=, >, !=, &&, ||), unary operators (!, -) and trigonometric operators (rsin, rcos, rtan, rasin, racos, ratan, sin, cos, tan, asin, acos, atan, dtor, rtod, sqrt, ln, exp). Additional operators (such as XOR and NAND) may be used in set directives but not in conditional expression clauses.

At least one set of parentheses must be used to enclose a conditional expression. Additional sets are recommended for proper evaluation of complex algorithms. Proper matching of parentheses is very important. If any parentheses are missing, a warning event message will be generated and the conditional expression will be evaluated as false.

Scenario-2.3.1.3 Simple Scenario Example

The following 4 lines are a simple example of a scenario file. The first line is a comment. The second line is a directive to the Scenario module to pause for one second. The third line is a directive to the linked module to change the value of the container item named “TransmitMode” to one. The fourth line starts execution of a nested scenario file named OtherScenario.txt.

```
;This is a comment line in a scenario file
SLEEP 1000
set TransmitMode 1
Start scenario D:/Scenario/OtherScenario.txt
```

Scenario-2.3.1.4 IF Scenario Example

```
if ((mnemonic0 + rsin(mnemonic1)) > mnemonic2)
     set mnemonic4 321
else if (!mnemonic3)
     set mnemonic5 20
     set mnemonic6 += 100
else
     set mnemonic7 512
endif
```

Scenario-2.3.1.5 While Scenario Example

The following is an example of a scenario file using a while statement. A sleep delay within the loop is necessary to prevent CPU-intensive wait loops from depleting system resources.

```
while ((mnemonic0 < 5)&& (mnemonic2 > sqrt(mnemonic3)))
     set mnemonic2 -= sqrt(mnemonic3)
     set mnemonic0 += 1
     Sleep 1000
endwhile
```

Scenario-3.0 Outputs

The Scenario module has 4 output channels. Each output channel should be linked to input channel 1 of the module to receive the directives.

Channel	Description
1-4	Scenario module sends the directives to the linked module.

Scenario-4.0 Container Items

The Scenario module accepts operator directives and is capable of receiving directives from other SIMSS modules. Use the **Set** and **Get** directives to access items with a fixed type. Use the **SetBuffer** and **GetBuffer** directives on buffer types. Although names in the following table contain upper and lower case, directive lines are not case-sensitive.

There are 6 container items for each of the five GUI-controlled scenarios. To modify or view most of these fields, it is easier to use the Run-time Control display. To access the items for the first controlled scenario, enter the name without the "<number>" field. To access the items for the 2nd, 3rd, 4th, and 5th controlled scenarios, replace the "<number>" field with "1", "2", "3", or "4". For example, to turn on echoing of the directive lines to the event log for the 5th controlled scenario, enter the following on the directive line with the Scenario module selected to receive it.

```
     set Scenario4DirMsg 1
```

To disable the echoing of directive lines to the log for the first controlled scenario, enter

```
     set ScenarioDirMsg 0
```

Name	Type	Description
Scenario<number>Currentline	Buffer	Current directive line in scenario file
Scenario<number>DirMsg	Fixed	Flag to list lines read from scenario file (0=no, 1=yes)
Scenario<number>Execution	Fixed	Run flag for scenario execution (0=pause, 1=run)
Scenario<number>Filename	Fixed	Scenario's Filename

Scenario< number >LineNumber	Fixed	Current line number in scenario file
Scenario< number >Status	Fixed	Scenario file's execution status (0=Finished, 1=Running, 99=Paused)

Scenario-5.0 Displays

To access displays for a module, click in the center of the module in the project window. The following module pop-up menu will appear. There are no configuration screens for the Scenario module. The Run-time option is available only when the project is running. The “Remove” option can be used during project design to remove the module. The “About” option is for the display of generic module information.

Module Pop-Up Menu Item	Description
Configure	Access Confguration menu for this module
Run-time	Access Run-time menu for this module
Remove	Remove module from the project
About	Display generic module information

Scenario-5.1 Configure

There are no configuration displays for this module. Refer to section Scenario-2.1 for information on configuring the scenario execution mode, *ConcurrentScenario*, from the Server/Properties/Property.txt file.

Scenario-5.2 Run-time

Click on the module pop-up menu “Run-time” option to request the Run-time menu.

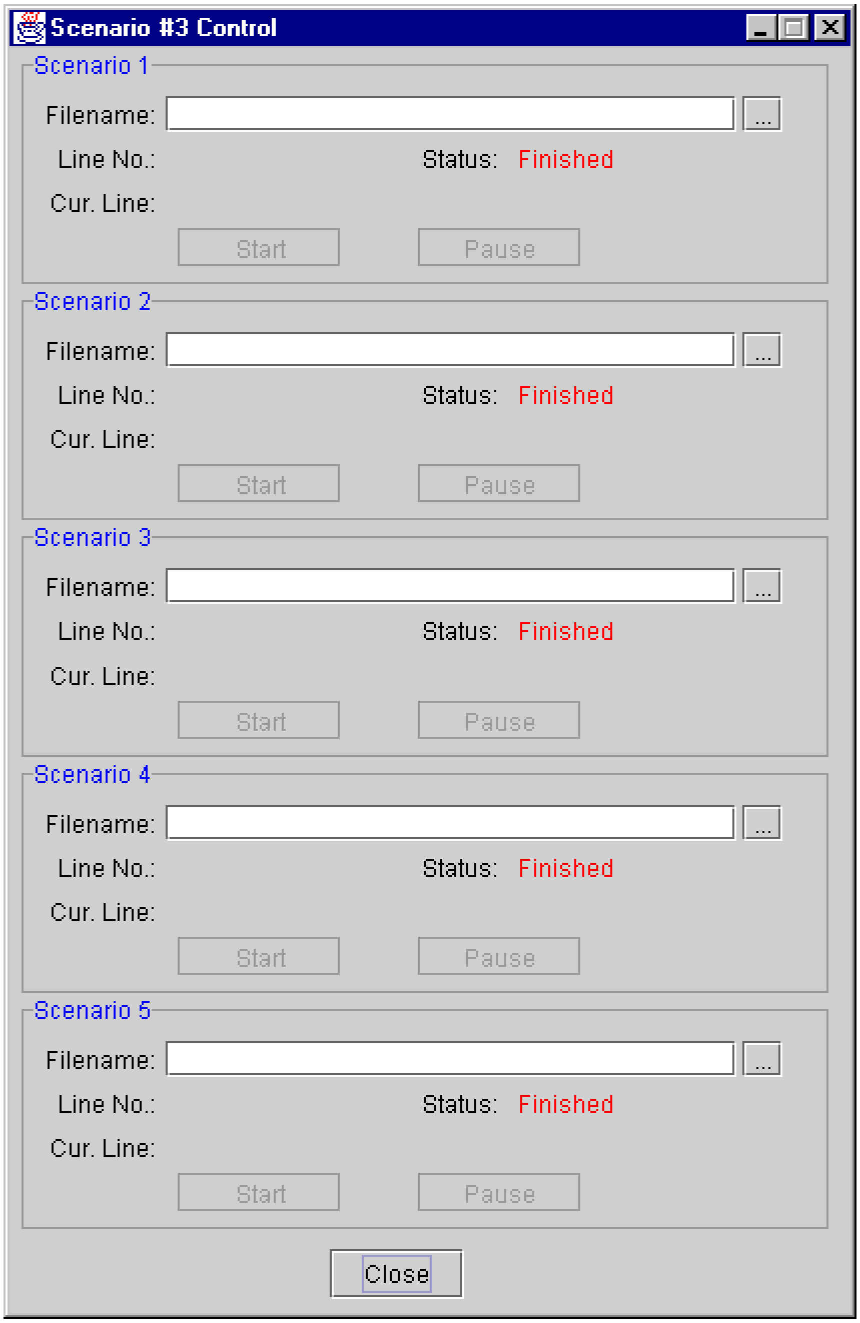
Run-time Menu Item	Description
Control	Request Control Scenario Display

Scenario-5.2.1 Scenario Control Display

Select “Control” from the Run-time menu to show the Scenario Control Display. Up to five scenario files may be controlled and monitored with this display. The fields associated with each file are described in the following table. Click the **Close** button to dismiss the display.

Field	Description
Filename	Enter the full or relative pathname for the scenario file in this field.
File browser ...	Click on this button to the right of the filename field to invoke file browsing to select a file.
Start/Stop Button	This button's title changes to Start or Stop as appropriate. To start processing an entered file, click its **Start** button. To stop an executing file, click its **Stop** button.
Pause/Resume Button	This button's title changes to Pause or Resume as appropriate. To temporarily stop the processing of a scenario file, click its **Pause**

	button. To resume a file after pausing, click its **Resume** button.
Line No.	This field shows the line position within the file.
Cur. Line	This field shows the text of the most recently executed line.
Status	This field shows “Running”, “Paused”, or “Finished”.



Scenario #3 Control

Scenario 1

Filename:

Line No.: Status: Finished

Cur. Line:

Scenario 2

Filename:

Line No.: Status: Finished

Cur. Line:

Scenario 3

Filename:

Line No.: Status: Finished

Cur. Line:

Scenario 4

Filename:

Line No.: Status: Finished

Cur. Line:

Scenario 5

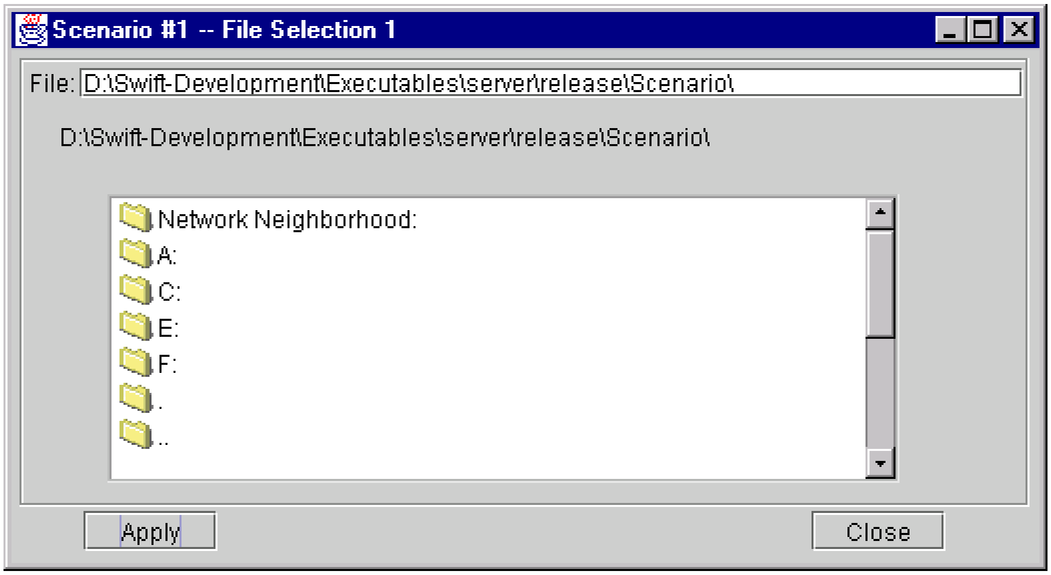
Filename:

Line No.: Status: Finished

Cur. Line:

Scenario-5.2.2 Scenario File Browser

When any of the file browser buttons is selected from the Scenario Control Display, a file selection display similar to the next screen will be displayed. The current search directory will be shown in the File field. Single click on the appropriate folder icons to navigate to the desired directory. Click on the ".." folder icon to go up the directory structure. Click on a named folder to display its contents. Single click to select a scenario file name. The file name should then appear in the File field. Use the **Apply** button to transfer the selected file name to the control display and dismiss the file selection window. Use the **Close** button to dismiss the display without selecting a file. When a file is selected in this manner, its directory then becomes the current search directory for the next file browse action.

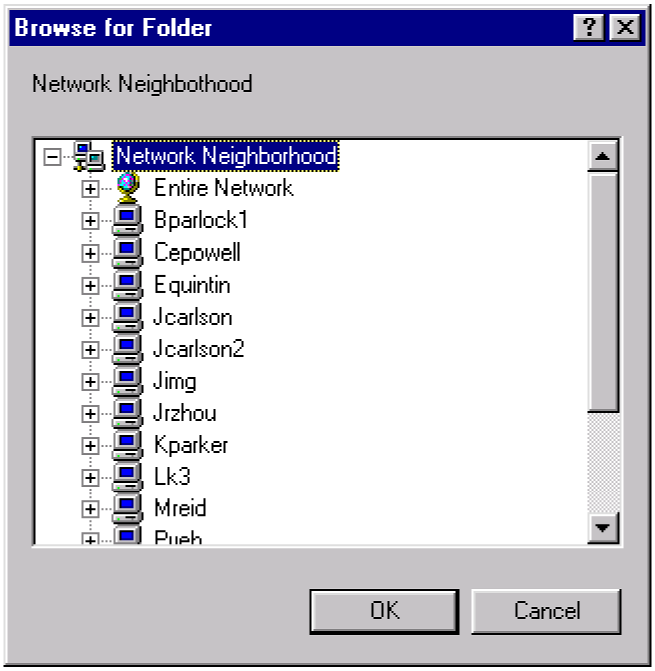


Scenario #1 -- File Selection 1

File: D:\Swift-Development\Executables\server\release\Scenario\

D:\Swift-Development\Executables\server\release\Scenario\

Note: This file browse window allows navigation to any folder within the network neighborhood. If the Network Neighborhood folder is clicked, another "Browse For Folder" window will be opened (**it may be hidden under the first browse window**).



Browse for Folder

Network Neighbothood

Navigate this browser to the desired folder and click the **OK** button or use the **Cancel** button to dismiss this browser.

If a network folder was selected, the files from that folder are now available for selection from the original file browse window. Scroll down to view these file names. Single click to select a file name. The file name should then appear in the File field. Use the **Apply** button to transfer the selected file name to the control display and dismiss the file selection window. Use the **Close** button to dismiss the display without selecting a file.

Scenario-5.3 About

Selecting the “About” option from the Scenario module pop-up menu produces a display that lists the module’s number of input and output channels, whether operator directives are allowed, names of authors, and the version number.

Scenario-6.0 Special Operating Instructions

Scenario-6.1 Built-in Delays in Scenario Files

Because of the fast processing of scenario files (e.g. 300 directives or more per second), the GUI Event Message window may become jammed by the volume of event messages produced in a very short time. Consequently, other displays may also be “hung up” by this undesirable effect. A proper remedy is under investigation. In the meantime it is suggested that delays be built into the scenario files so that no more than 20 directives are executed each second. If scenario event messages are being filtered out from the event window, up to 50 directives may be executed each second.

```
; Sample scenario file
Set MOD_SS_DRNADSTEP 1
SET CEF_NR_ETRQ_1 2
SET AS2_TS_DETECTOR 3
SLEEP 1000
SET AMR_NR_MWASPEED 65
SET AMR_SS_BITPKT1 1
SET AMR_SS_BITPKT4 1
SLEEP 125
SET EPS_SS_MAINBUSVR 0XFF
sleep 5000
SETBUFFER TLMPACKET0140 20 BYTE 0XA 10 012
```

Scenario-6.2 Stopping Nested or Triggered Scenarios

Even though the user does not have control over the execution of scenarios started internally, it is possible to stop them without shutting down the simulator. To stop all ongoing scenario files of all types, click on the Project's Run menu and select the "Stop" option. The user should then immediately click on the "Run" option to restart the entire project. Since all of the modules will be stopped briefly, there could be dropouts in data and other consequences. The scenario module does not automatically restart any scenario files when restarted after a project stop.

Serial Input (SI) Module

SI-1.0 Overview

The Serial Input (SI) module provides the capability to receive serial data through a port on an ICS serial card in the host computer and to pass the data received to other modules.

SI-2.0 Input

The SI module does not have any input channels.

SI-3.0 Output

Channel	Description
1	Data received from the serial port is passed on this channel

SI-4.0 Container Items

The SI module's container items are not accessible via operator directives, so they are not listed here.

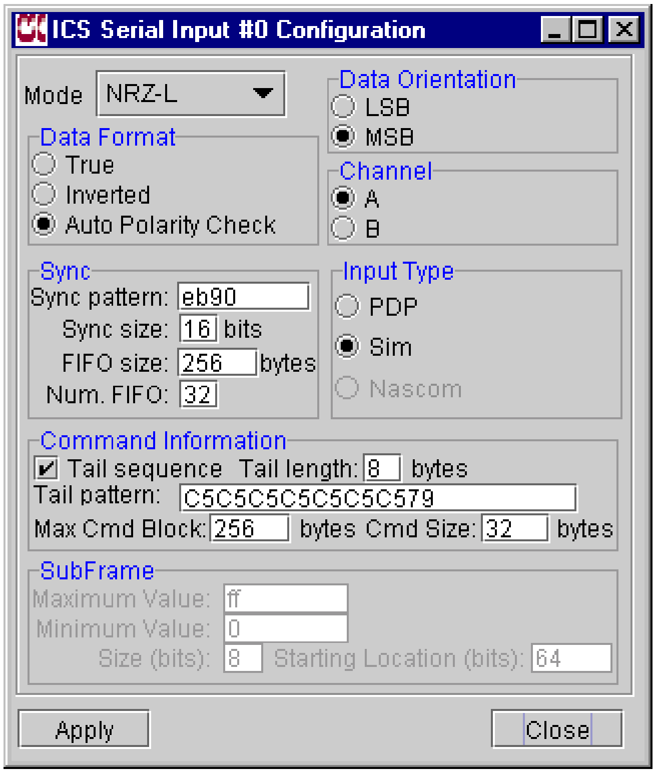
SI-5.0 Displays

To access displays for this module, click in the center of the SI module in the project window. The following module pop-up menu choices will appear. The "Configure" and "Remove" options are only available during project design. The "Run-time" option is only available when the project is running.

Module Pop-up Menu Item	Description
Configure	Access the Serial Input configuration window
Run-time	Access the module's run-time menu
Remove	Remove the module from the project
About	Display generic module information

SI-5.1 Configuration

Select the "Configure" item from the module pop-up menu to access the configuration window for the Serial Input module.



ICS Serial Input #0 Configuration

Mode: NRZ-L

Data Orientation
○ LSB
● MSB

Data Format
○ True
○ Inverted
● Auto Polarity Check

Channel
● A
○ B

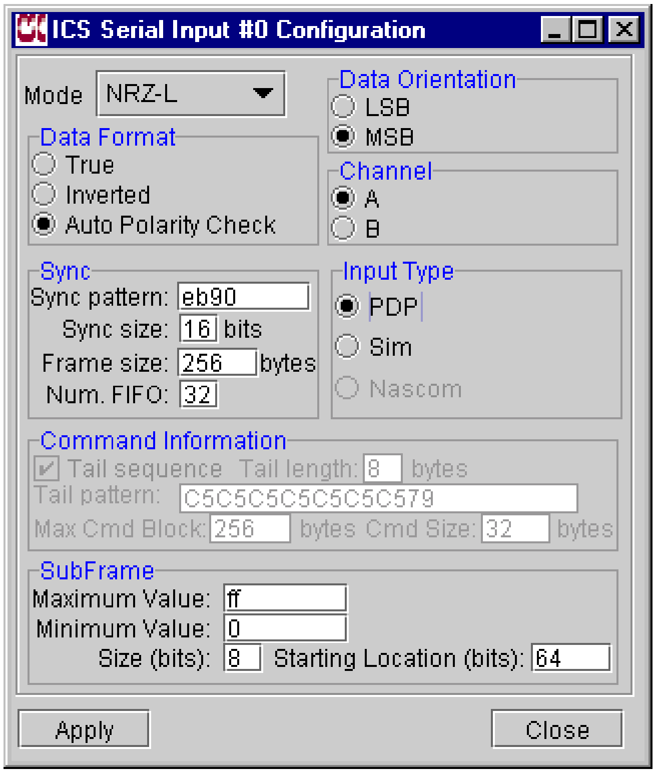
Sync
Sync pattern: eb90
Sync size: 16 bits
FIFO size: 256 bytes
Num. FIFO: 32

Input Type
○ PDP
● Sim
○ Nascom

Command Information
☑ Tail sequence Tail length: 8 bytes
Tail pattern: C5C5C5C5C5C5C579
Max Cmd Block: 256 bytes Cmd Size: 32 bytes

SubFrame
Maximum Value: ff
Minimum Value: 0
Size (bits): 8 Starting Location (bits): 64

The default configuration is for the Sim Input Type. When the Input Type field is set to Sim, the Command Information section is enabled for entry and the SubFrame section is disabled.



ICS Serial Input #0 Configuration

Mode: NRZ-L

Data Orientation
○ LSB
● MSB

Data Format
○ True
○ Inverted
● Auto Polarity Check

Channel
● A
○ B

Sync
Sync pattern: eb90
Sync size: 16 bits
Frame size: 256 bytes
Num. FIFO: 32

Input Type
● PDP
○ Sim
○ Nascom

Command Information
☑ Tail sequence Tail length: 8 bytes
Tail pattern: C5C5C5C5C5C5C579
Max Cmd Block: 256 bytes Cmd Size: 32 bytes

SubFrame
Maximum Value: ff
Minimum Value: 0
Size (bits): 8 Starting Location (bits): 64

When “PDP” is selected as the Input Type, the “SubFrame” options will be enabled, “Command Information” will be disabled, and “Frame Size” replaces “FIFO Size” within the Sync section of the display.

Parameter	Description
Mode	Select the Pulse Code Modulation (PCM) decoding method of the input stream from NRZ-L**,** NRZ-M**,** or NRZ-S
Data Orientation	Select "LSB" if the data stream is received with the least significant bit first. Select "MSB" if the data stream is received with the most significant bit first.
Data Format	Select "True" if the polarity of the data stream is normal. Select "Inverted" if the polarity of the data stream is inverted. Select "Auto Polarity Check (APC)" if the polarity is unknown.
Channel	Select "Channel A" or "Channel B" for the operation. "Channel A" refers to the first ICS board installed in the system, while "Channel B" refers to the second one.
Sync Pattern	Enter the synchronization pattern in hexadecimal format for the serial I/O card to receive data (maximum 8 digits).
Sync Size	Enter the size of the synchronization pattern in bits (maximum 32 bits)
Frame Size	Enter number of bytes per frame to be received (maximum 4096 bytes).
FIFO Size or Frame Size	Enter the FIFO' s size (maximum 4096) in "SIM" mode. Enter the data frame size (maximum 4096 bytes) in "PDP" mode
Num. FIFO	Enter number of FIFO to receive data (maximum 99)
Input Type	Select "PDP" if the input stream is telemetry. Select "SIM" if the input stream is command.
Tail Sequence	If the command bit stream has a tail sequence, check this box.
Tail Length	Enter the command tail sequence size in bytes (maximum 8 bytes)
Tail Pattern	Enter the command tail sequence pattern in hexadecimal digits (maximum 16 digits).
Max Cmd Size	Enter the maximum command size expected in bytes (maximum 4096 bytes)
Cmd Size	Enter command size in bytes (1 to Max Cmd Size). Serial Input module will separately pass to next module sync patterns, bytes of command data (specified by "Cmd Size"), and post-amble. If Cmd Size equals Max Cmd Size, Serial Input module will pass to next module sync patterns, command(s) data, and post-amble in one packet.
Maximum Value	Enter telemetry subframe's maximum value.
Minimum Value	Enter telemetry subframe's minimum value.
Size	Enter telemetry subframe size in bits (maximum 16 bits)
Starting Location	Enter the starting location for the telemetry subframe in bits (greater than 32)

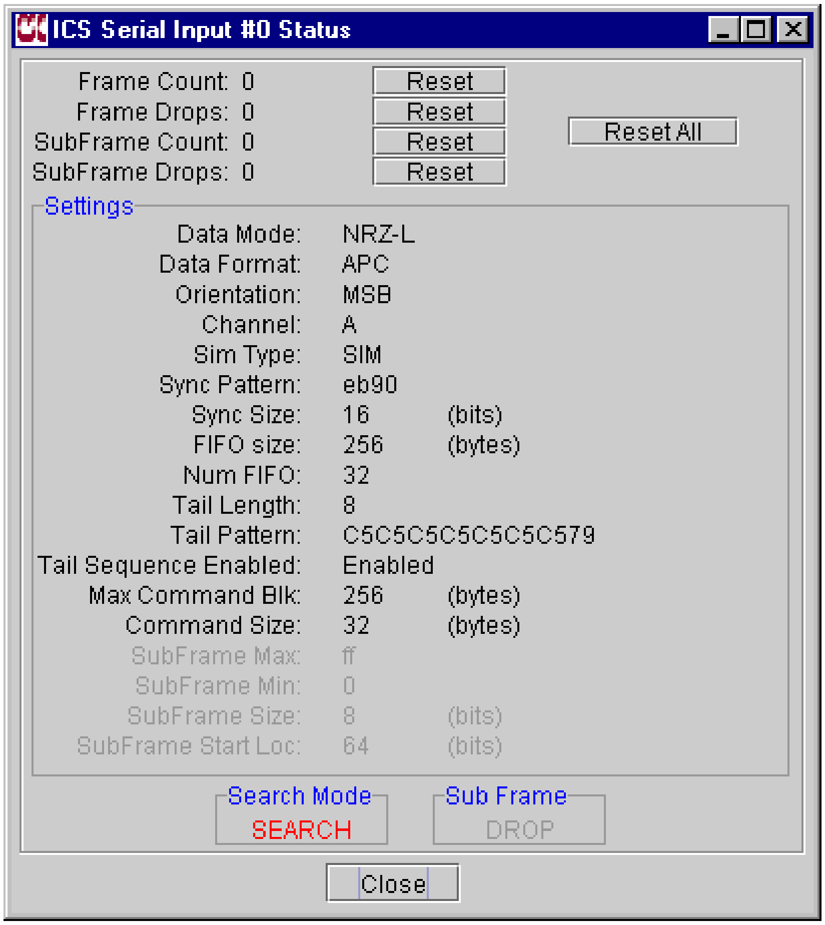
Click **Apply** button to configure any changes. Click **Close** button to dismiss the display.

SI-5.2 Run-time

Click the module pop-up menu "Run-time" option to access the Run-time menu with the following choices.

Run-time Menu Item	Description
Display Status	Show the Status window of the Serial Input module
Display Dump	Show the Dump formats available
Resume	Restart the module
Pause	Stop the module

SI-5.2.1 Display Status



Frame Count: 0
Frame Drops: 0
SubFrame Count: 0
SubFrame Drops: 0

Data Mode:	NRZ-L	
Data Format:	APC	
Orientation:	MSB	
Channel:	A	
Sim Type:	SIM	
Sync Pattern:	eb90	
Sync Size:	16	(bits)
FIFO size:	256	(bytes)
Num FIFO:	32	
Tail Length:	8	
Tail Pattern:	C5C5C5C5C5C5C579	
Tail Sequence Enabled:	Enabled	
Max Command Blk:	256	(bytes)
Command Size:	32	(bytes)
SubFrame Max:	ff	
SubFrame Min:	0	
SubFrame Size:	8	(bits)
SubFrame Start Loc:	64	(bits)

Search Mode: SEARCH
Sub Frame: DROP

This display shows the number of frames and sub-frames that have been received and dropped. It also shows the current configuration settings. Please refer back to the table in section SI-5.1 for descriptions of the settings.

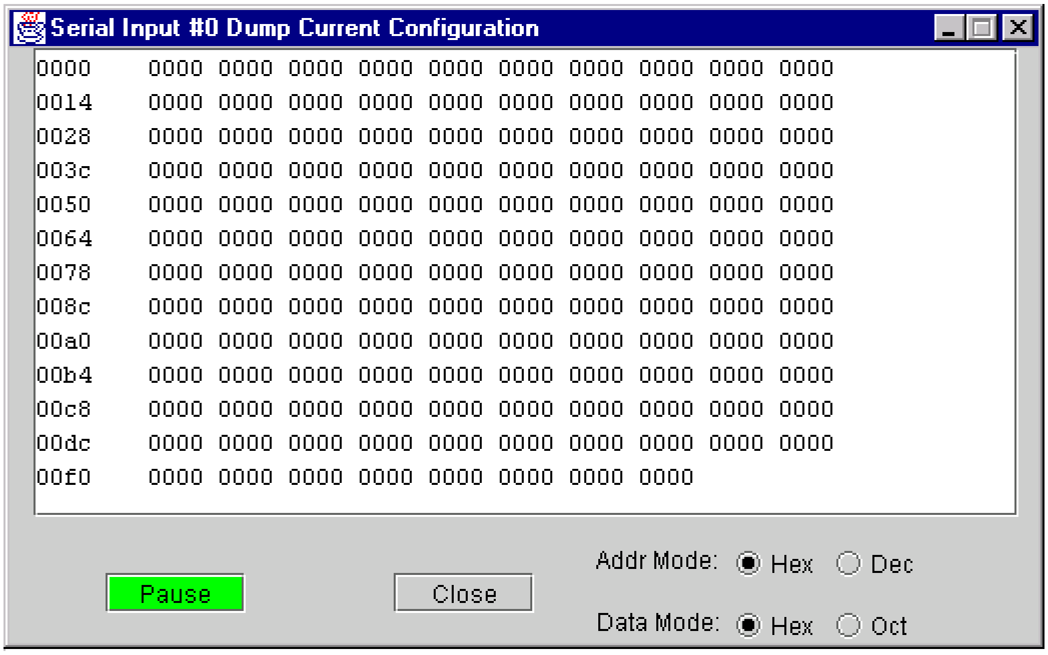
Search Mode and Sub Frame status fields are only applicable when receiving telemetry data.

Click any **Reset** button to change its associated counter to zero.

Click the **Close** button to dismiss the display.

SI-5.2.2 Display Dump

Select the “Display Dump” option to view the available dump options. Currently there is only one format available. When this option is selected the data received from the serial I/O port is displayed.



```
0000     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0014     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0028     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
003c     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0050     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0064     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0078     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
008c     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
00a0     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
00b4     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
00c8     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
00dc     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
00f0     0000 0000 0000 0000 0000 0000 0000 0000
```

Addr Mode: ● Hex ○ Dec

Data Mode: ● Hex ○ Oct

Button	Description
Addr Mode	Select “Hex” to display the memory address in hexadecimal format Select “Dec” to display the memory address in decimal format
Data Mode	Select “Hex” to display data in hexadecimal format Select “Oct” to display data in octal format
Pause	Pause updating the data contained in the window. The button name changes to **Cont**.
Cont	Continue updating a previously paused display. The button name changes to **Pause**.
Close	Close the window

SI-5.2.3 Resume

After an individual module's execution has been paused, select “Resume” from the Run-time menu to restart the module. The color around the module’s border will change from red (indicates stopped state) to green (indicates run state).

SI-5.2.4 Pause

Select the “Pause” option from the Run-time menu to temporarily stop the Serial Input module’s processing. The color around the module’s border will change from green (indicates run state) to red (indicates stopped state). The Serial Input module can be re-configured by selecting the “Configure” option of the module pop-up menu. To return to run mode, select the “Resume” option.

SI-5.3 Remove

Select this option to remove the Serial Input module from the project.

SI-5.4 About

Selecting the “About” option from the module pop-up menu produces a display that lists the number of input channels, number of output channels, whether directives are allowed, names of authors and the module’s version number.

SI-6.0 Special Operating Instructions

“Max Cmd Block” and “Cmd Size” Configuration

In “Sim” mode, Serial Input module has two different ways to pass received command data to next module, depending on how “Max Cmc Block” and “Cmd Size” are configured. “Max Cmd Block” is the maximum size of a block of command data including the sync patterns and tail sequence. “Cmd Size” is the size of each command within a block of command data. A block of command has from one to many commands.

1/ If a block of command data has multiple commands and user wants individual command to be passed to next module whenever it is fully received, then the “Cmd Size” should be set to exactly the size of one command. In this case, Serial Input module will pass sync patterns, first command, second command, third command,…, and finally tail sequence to next module in separate packets.

2/ If user wants the whole block of command data from sync patterns to tail sequence to be passed to next module in one packet (no matter how many commands included within the block), then the “Cmd Size” must be equal to the “Max Cmd Block”. In this case, Serial Input module will pass to next module the whole block of data from sync patterns to tail sequence or to “Max Cmd Block” bytes, whichever comes first.

Serial Output (SO) Module

SO-1.0 Overview

The Serial Output (SO) module provides the capability to transmit serial data through ports on an ICS serial card in the host computer.

SO-2.0 Input

Channel	Description
1	Receive data from upper module through this channel

SO-3.0 Output

Channel	Description
1	Pass transmitted data to next module through this channel

SO-4.0 Container Items

The Serial Output module's container items allow for changing the output frequency while the project is running. It is suggested that data transmission be halted prior to changing the output frequency to prevent loss of data.

Name	Type	Description
SerialFrequency	Fixed	Clock frequency in hertz at which the data will be transmitted.
SerialZZChannel	Fixed	Set to re-enable output after changing the transmit frequency. The argument is zero for channel A and one for channel B. NOTE: This does not re-enable output from the upstream module.

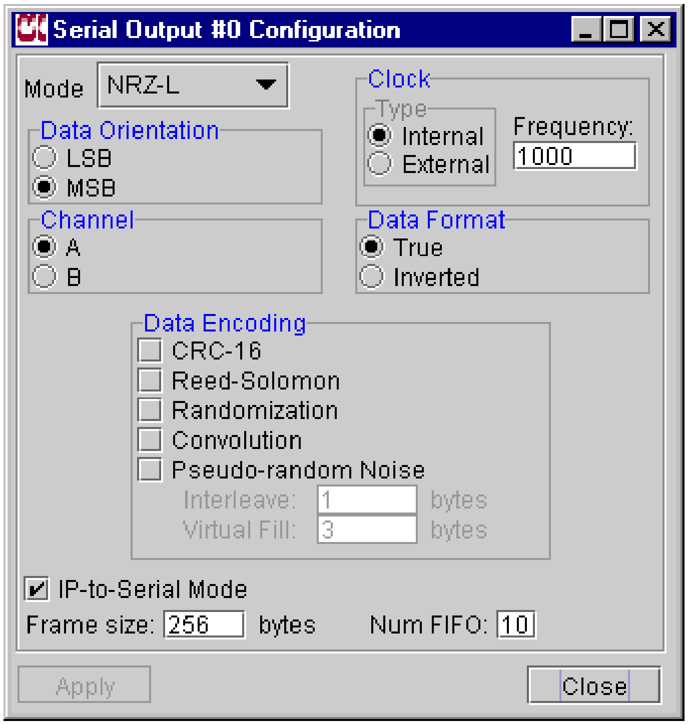
SO-5.0 Displays

To access the displays for this module, click in the center of the module in the project window. The module pop-up menu will appear with the following choices. The "Configure" and "Remove" options are only available during project design. The "Run-time" option is only available when the project is running.

Module Pop-up Menu Item	Description
Configure	Access the Serial Output configuration window
Run-time	Access to module's Run-time menu
Remove	Remove the module from the project
About	Display generic module information

SO-5.1 Configuration

Select the “Configure” item of the module pop-up menu to access the configuration window of the Serial Output module.



Serial Output #0 Configuration

Mode: NRZ-L

Data Orientation
○ LSB
● MSB

Channel
● A
○ B

Clock
Type
● Internal
○ External
Frequency: 1000

Data Format
● True
○ Inverted

Data Encoding
□ CRC-16
□ Reed-Solomon
□ Randomization
□ Convolution
□ Pseudo-random Noise
Interleave: 1 bytes
Virtual Fill: 3 bytes

☑ IP-to-Serial Mode
Frame size: 256 bytes   Num FIFO: 10

Parameter	Description
Mode	Select the PCM encoding method of the data stream from NRZ-L, NRZ-M, NRZ-S, BIO-L, BIO-M, or BIO-S.
Clock Type	In this version, only the internal clock is used.
Clock Frequency	Enter the frequency in Hz to transmit data out.
Data Orientation	Select “LSB” if the data stream is transmitted with the least significant bit first. Select “MSB” if the data stream is transmitted with the most significant bit first
Channel	Select “Channel A” or “Channel B” for the operation. “Channel A” refers to the first ICS board installed in the system while “Channel B” refers to the second one.
Data Format	Select “True” if the polarity of the data stream is normal. Select “Inverted” if the polarity of the data stream is inverted.
Data Encoding	Click the appropriate boxes to enable “CRC-16”, “Reed-Solomon”, “Randomization”, “Convolution” and “Pseudo-random Noise” encoding.
Interleave	If “Reed-Solomon” encoding is enabled, enter the length in bytes for interleave.
Virtual Fill	If “Reed-Solomon” encoding is enabled, enter number of bytes for virtual fill.
IP-to-Serial Mode	Check this option if the Serial Output module is configured to receive data from an Input IP module.

Frame Size	Enter total number of bytes per frame to be transmitted (maximum 4096 bytes)
Num. FIFO	Enter number of FIFO to contain transmitting data (maximum 99)

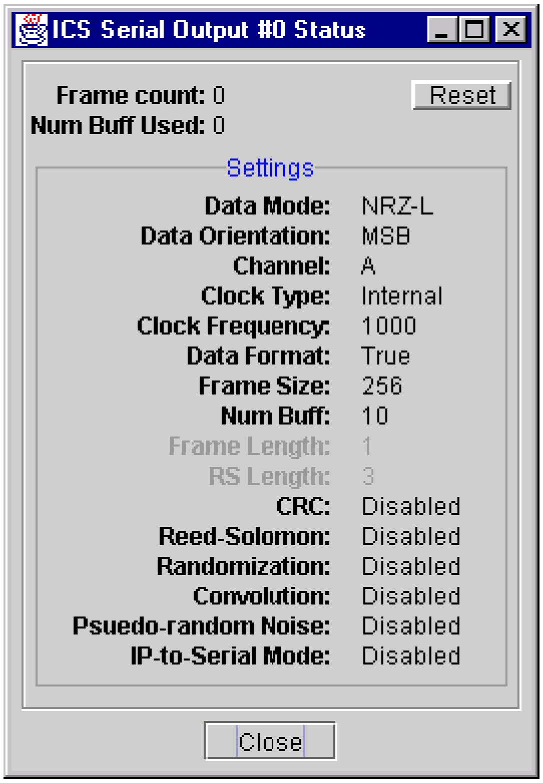
Click on **Apply** button to configure any changes. Click on **Close** to dismiss the display.

SO-5.2 Run-time

Click on the module pop-up menu “Run-time” option to request the Run-time menu.

Run-time Menu Item	Description
Display Status	Show the status window of the Serial Output module
Display Dump Menu	Show options for displaying data within a frame
Resume	Restart the module
Pause	Stop the module

SO-5.2.1 Display Status



Frame count: 0
Num Buff Used: 0

Data Mode: NRZ-L
Data Orientation: MSB
Channel: A
Clock Type: Internal
Clock Frequency: 1000
Data Format: True
Frame Size: 256
Num Buff: 10
Frame Length: 1
RS Length: 3
CRC: Disabled
Reed-Solomon: Disabled
Randomization: Disabled
Convolution: Disabled
Psuedo-random Noise: Disabled
IP-to-Serial Mode: Disabled

This display shows the current frame counter, number of FIFO used, and the currently configured settings. Please refer to the table in section SerialOutput-5.1 for descriptions of the settings.

Click **Reset** to clear the frame counter to zero.

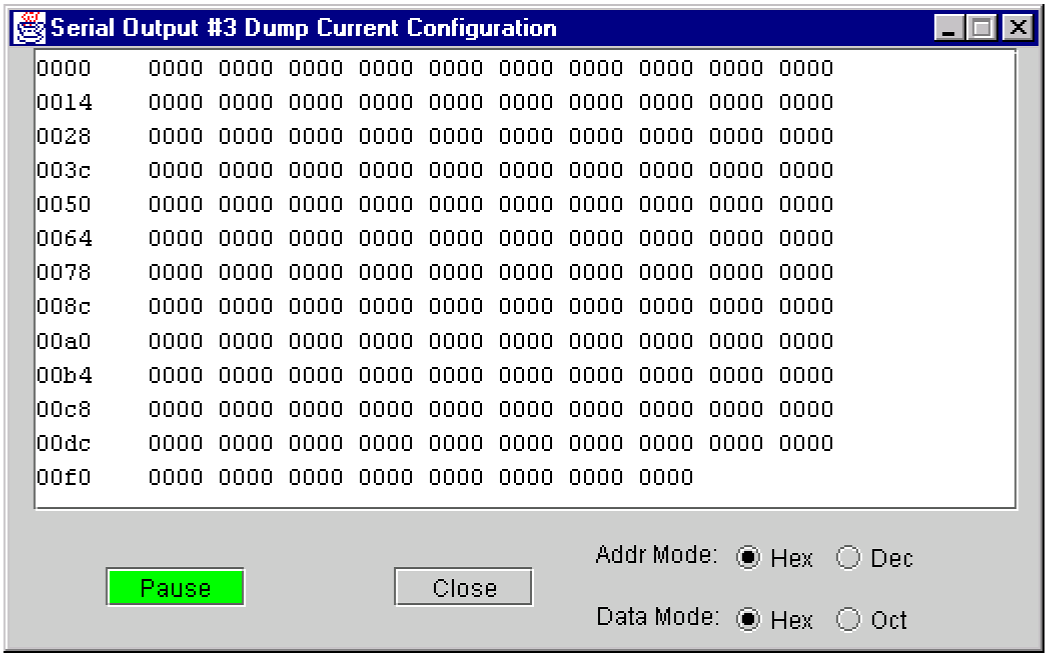
Click **Close** to close the window.

Note: if the Serial Output module is configured to receive data from an Input IP module, the “Num FIFO Used” value varies from 0 to “Num FIFO” value, defending on the difference of data rate between Input IP module and Serial Output module. In any other configuration the “Num FIFO Used” is always at “Num FIFO” value.

SO-5.2.2 Display Dump

Select the “Display Dump” option to view the available dump formats. Currently there is only one format available. When this option is selected, the output data buffer is displayed. This display has the following standard dump buttons.

Buttons	Description
Addr Mode	Select “Hex” to display the addresses in hexadecimal format Select “Dec” to display the addresses in decimal format
Data Mode	Select “Hex” option to display data in hexadecimal format Select “Oct” option to display data in octal format
Pause	Freeze the data window contents. The **Pause** button is renamed **Cont**.
Cont	Resume updating the data window contents. The **Cont** button is renamed **Pause**.
Close	Close the window



Serial Output #3 Dump Current Configuration

```
0000    0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0014    0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0028    0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
003c    0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0050    0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0064    0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0078    0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
008c    0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
00a0    0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
00b4    0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
00c8    0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
00dc    0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
00f0    0000 0000 0000 0000 0000 0000 0000 0000
```

Addr Mode: ● Hex ○ Dec

Data Mode: ● Hex ○ Oct

SO-5.2.3 Resume

After an individual module has been stopped, select “Resume” from the Run-time menu to restart the module. The color around the module’s border will change from red (indicates stopped state) to green (indicating it has entered the run state).

SO-5.2.4 Pause

Select “Pause” from the Run-time menu to temporarily stop the Serial Output module’s processing. The color around the module’s border will change from green (indicates run state) to red (indicates stopped state). While Paused, the Serial Output module can be re-configured by selecting the “Configure” option of the module pop-up menu. To return to run mode, select “Resume” from the Run-time menu.

SO-5.3 Remove

Select this option to remove the Serial Output module from the project.

SO-5.4 About

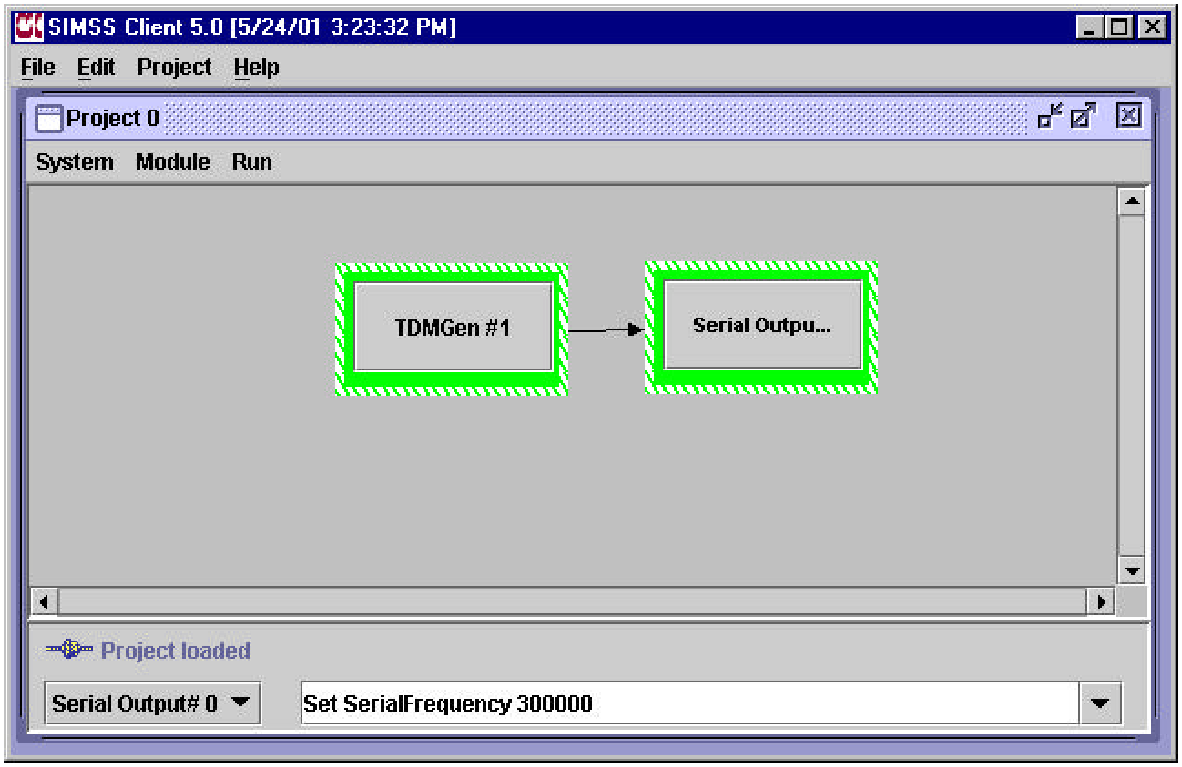
Selecting the “About” option from the module pop-up menu produces a display that lists the number of input channels, number of output channels, whether directives are allowed, names of authors and the module’s version number.

SO-6.0 Special Operating Instructions

Operation with Input IP module

When the Serial Output module is configured to receive data from an Input IP module, the “IP-to-Serial Mode” option should be checked. When the project is running, bring up the Status window of the Serial Output module to monitor the “Num FIFO Used”, and look at the event messages to know the input data rate of IP data. By looking at “Num FIFO Used”, buffer under flow or buffer over flow can be detected. It is necessary to set the data rate of Serial Outp ut module about the same as the data rate of the IP data. See “Setting Frequency on the Fly” in this section for the instructions on how to set a new frequency during run-time, or manually stop the Serial Output module to set a new frequency and re-start it again.

Setting Frequency on the Fly



Project loaded

Serial Output# 0

Set SerialFrequency 300000

The directive window of the project GUI can be used to change the serial output frequency during run-time. Follow the steps given below.

1. Click on the drop-down menu on the bottom left of the directive window to select the Serial Output module for which to change the frequency value

2. In the directive input field, type “Set SerialFrequency ” + integer value in Hz, then press the <Enter> key. For example, to set the new frequency to 300000 Hz:

   Set SerialFrequency 300000 + <Enter>

3. Type “Set SerialZZChannel ” + current channel value in integer (A = 0, B = 1), then press the <Enter> key. For example, if the current channel is channel B:

   Set SerialZZChannel 1 + <Enter>

4. Check that the serial output channel is operating at the new frequency.

Transmit File (TxFile) Module

TxFile-1.0 Overview

The Transmit File (TxFile) module reads in data from a file according to user-defined parameters and sends the data out unformatted. It operates in either manual mode, where data blocks are loaded and sent out one at a time under user control, or automatic mode, where data blocks are loaded and sent out under user-defined parameters. TxFile can transmit the data in IP transmission mode or in serial transmission mode. TxFile can process a file in normal order or reversed order (the first byte of the file is read and sent out as last byte, the last byte of the file is read and sent out as first byte). If the data file was created by the Log Module and has log headers, it is possible to retransmit the file with the same relative timing between blocks.

TxFile-2.0 Inputs

The TxFile module does not technically have any input channels. It is not possible to create an input link to this module. Its lone output channel is actually bi-directional, providing the down-linked module an optional method for controlling the flow of output data. This method is only used when the TxFile is connected to a module requesting serial transmission of data.

TxFile-3.0 Outputs

Channel	Description
1	Data read from file

TxFile-4.0 Container Items

This module accepts operator directives and is capable of receiving directives from a Scenario module. Use the Set and Get directives to access items with a fixed, integer, float, or string types. Use the SetBuffer and GetBuffer directives on buffer types. Although names in the following tables contain upper and lower case, directive lines are not case sensitive.

Name	Type	Description
TxFileName	String	Pathname of file to send
TxFileStart	Integer	The position of the first block begin to read
TxFileSize	Fixed	Size of file in bytes
TxFilePosition	Integer	Current file pointer position
TxFileTransmitCount	Integer	Number of blocks transmitted
TxFileTransmitBuffer	Buffer	Most recent buffer loaded from file
TxFileBlockSize	Integer	Size of block to load and then transmit
TxFileReversed	Integer	Reversed order flag
TxFileMode	Integer	File Read Modes: (1) Offset (2) Log file (playback a logged file with header) (3) Sync (4) Length **Note:** (3) & (4) not implemented yet

TxFileModeOffset	Integer	Offset from the start position of previous block
TxFileLogTiming	Integer	Use log file timing instead of the interval
TxFileAuto	Integer	Automatic or manual mode flag
TxFileLoad	Integer	Flag to load a block
TxFileFileLoopFlag	Integer	Flag for number of times a whole file to be read and sent out
TxFileLoopCount	Integer	Number of times a whole file to be read and sent out. **Note:** a whole file would be read and sent out indefinite times if the counter is set to be equal or less than 0.
TxFileBlockFlag	Integer	Flag for number of blocks to be read and sent out
TxFileReadCount	Integer	Number of blocks to be read
TxFileSendCount	Integer	Number of times each block to be sent out
TxFileInterval	Float	Interval between two blocks being sent in milliseconds
TxFileSend	Integer	Flag to start transmission
TxFileContinueSend	Integer	Flag to continuously transmit
TxFileStop	Integer	Flag to stop transmission
TxFileTransmitMode	Integer	Output transmission mode (0=IP (default), 1=serial, 2=force IP mode)
TxFileSerialStatus	Integer	Serial transmission status (0=stopped automatically, 1=stopped by operator, 2=running, 3=waiting for send key)

TxFile-5.0 Displays

To access the displays for this module, first click on the center of the module in the project window. The following items will appear in a pop-up menu.

Module Pop-Up Menu Item	**Description**
Configure	Access the configuration menu for the module
Run-time	Access the Run-time menu for the module
Remove	Remove module from the project
About	Display generic module information

TxFile-5.1 Configuration Displays

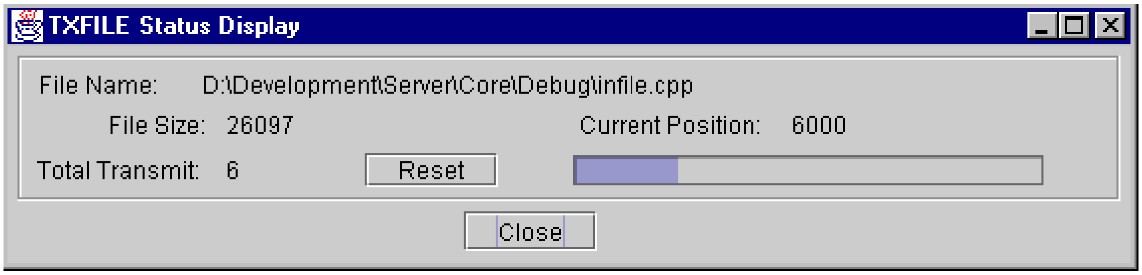
There are no configuration displays for this module. However, if during run-time TxFile receives a request for more data signal from the down-linked module, the transmission mode will be automatically reconfigured to serial and an event message will be generated to inform the operator.

TxFile-5.2 Run-time Displays

Run-time Menu Item	**Description**
Display Status	Display TxFile module status
Display Transmit Buffer	Display current block for transmission
Send From File	Control TxFile module processing

TxFile-5.2.1 Status Display

The status display shows the current filename, the size of the file, current file pointer position, and the total number of blocks transmitted.



TXFILE Status Display

File Name: D:\Development\Server\Core\Debug\infile.cpp

File Size: 26097

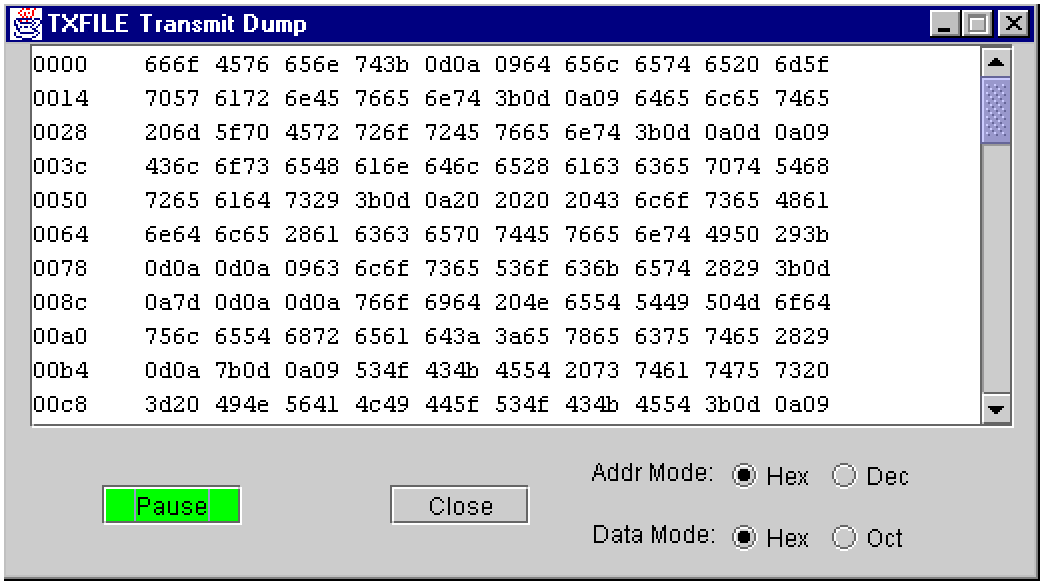
Current Position: 6000

Total Transmit: 6

Status Display Field	Description
File Name	Pathname of the file being transmitted
File Size	Size (in bytes) of the file being transmitted
Current Position	Current byte position in file
Total Transmit	Total number of blocks transmitted
Reset	Click to clear the **Total Transmit** count

TxFile-5.2.2 Transmit Buffer Display

The transmit buffer display shows the contents of the currently loaded block.



TXFILE Transmit Dump

```
0000     666f 4576 656e 743b 0d0a 0964 656c 6574 6520 6d5f
0014     7057 6172 6e45 7665 6e74 3b0d 0a09 6465 6c65 7465
0028     206d 5f70 4572 726f 7245 7665 6e74 3b0d 0a0d 0a09
003c     436c 6f73 6548 616e 646c 6528 6163 6365 7074 5468
0050     7265 6164 7329 3b0d 0a20 2020 2043 6c6f 7365 4861
0064     6e64 6c65 2861 6363 6570 7445 7665 6e74 4950 293b
0078     0d0a 0d0a 0963 6c6f 7365 536f 636b 6574 2829 3b0d
008c     0a7d 0d0a 0d0a 766f 6964 204e 6554 5449 504d 6f64
00a0     756c 6554 6872 6561 643a 3a65 7865 6375 7465 2829
00b4     0d0a 7b0d 0a09 534f 434b 4554 2073 7461 7475 7320
00c8     3d20 494e 5641 4c49 445f 534f 434b 4554 3b0d 0a09
```

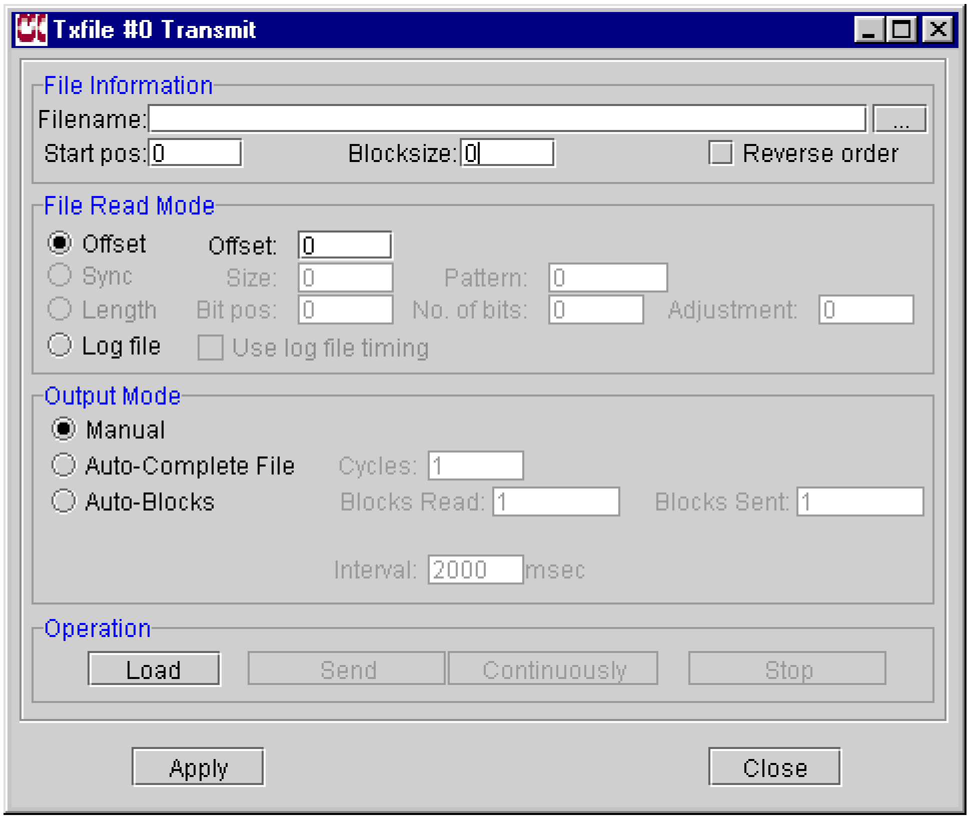
Addr Mode: ● Hex ○ Dec

Data Mode: ● Hex ○ Oct

The address field of the dump display may be toggled between hexadecimal and decimal display formats. The data portion may be toggled between hexadecimal and octal formats. The screen contents may be "frozen" using the **Pause** button. Paused data may be "unfrozen" with the **Cont** button.

TxFile-5.2.3 Send From File

The send from file display is the main operations display. This display allows the user to specify data files, specify read mode parameters, specify output mode parameters and control the data transmission. The following description mainly applies to the default IP transmission mode. For a description of this display's operation from serial transmission mode, see section TxFile-5.2.3.7 Example 5 Serial Transmission Mode.



File Information

Filename:

Start pos: 0   Blocksize: 0   □ Reverse order

File Read Mode

● Offset   Offset: 0

○ Sync   Size: 0   Pattern: 0

○ Length   Bit pos: 0   No. of bits: 0   Adjustment: 0

○ Log file   □ Use log file timing

Output Mode

● Manual

○ Auto-Complete File   Cycles: 1

○ Auto-Blocks   Blocks Read: 1   Blocks Sent: 1

Interval: 2000 msec

Operation

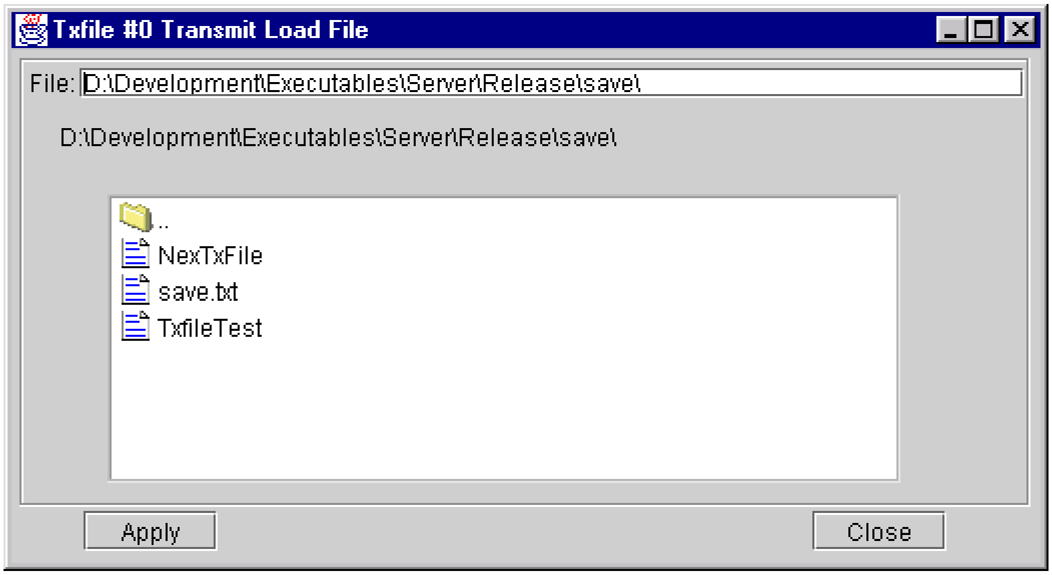
This is the initial Send from File Display before any parameters have been set. The default interval (for automatic transmission) is 2000 milliseconds (2 seconds) between blocks. The default read mode is Offset. The default output mode is Manual.

File Information	Description
Filename	Pathname of the file to transmit can be relative or absolute (leading or trailing white space is allowed, but not embedded spaces).
File browser ...	Click on this button to the right of the filename field to invoke file browsing to select a file.
Start pos	Starting position (in bytes) within the file of the first block to send.
Blocksize	Size of block to read (in bytes).
Reverse Order	If set, the file is transmitted in reversed order (the first bit of file is transmitted as the last bit, last bit of file is transmitted as first bit).
File Read Mode	**Description**
Offset button	Click to select Offset read mode
Offset field	Increment (in bytes) between the start of consecutive blocks. Should normally be the same as the block size. If zero, the same block would be read and sent out repeatedly.

Sync	**(Not yet implemented)** Sync mode -- Look for a sync pattern in the file as given by the bit size and the bit pattern and send out the number of bytes given by block size starting with and including the sync pattern.
Length	**(Not yet implemented)** Length mode -- Look for a length value in the file as given by the bit offset after the end of the previous block. Pick up the number of bits given, add the adjustment, and use that as the length of the block to send.
Log file	Specify a Log Module file with log headers
Use log file timing	Use relative timing from the log headers instead of the **Interval** to transmit data blocks
Output Mode	**Description**
Manual	(Manual mode) Data blocks are loaded and sent out one at a time under user control
Auto-Complete File	(Automatic mode) Transmit a whole file the specified number of **Cycles.**
Cycles	To transmit the file a fixed number of times, set **Cycles** to a nonzero value (default value is 1). To transmit the file an indefinite number of times, set **Cycles** to zero. **Note:** If the “Offset field" is set to zero, the end of file is never reached and the same block will be transmitted repeatedly until stopped.
Auto-Blocks	(Automatic mode) Transmit a portion of a file, indicated by setting **Blocks Read** and **Blocks Sent** (both are 1 by default).
Blocks Read	Number of blocks to read
Blocks Sent	Number of times each read block is sent out. **Note:** If this field is less than or equal to 0, the blocks read are sent out repeatedly until stopped.
Interval	Interval between block transmission (in milliseconds).
Operation	**Description**
Load	**(Manual mode only)** Loads the next block from the file.
Send	**Manual mode**: Sends the currently loaded block one time. **Automatic modes**: Starts the specified automatic transmission.
Continuously	**(Auto-Blocks mode only)** Pressing the **Send** button sends out the current block. Pressing the **Continuously** button will make the next block the current block and send it out.
Stop	**(Automatic modes only)** Stop sending blocks.
Apply	Configuration values are used only after this button is pressed**.**
Close	Close this window.

TxFile-5.2.3.1 File Browsing

When the file browser button is selected from the TxFile Transmit Display, a file selection display similar to the next screen will be displayed. The current search directory will be shown in the File field. Double click on the appropriate folder icons to navigate to the desired directory. Double click on the ".." folder to go up the directory structure. Double click on a named folder to display its contents. Double click to select a scenario file name. The file name should then appear in the File field. Use the **Apply** button to transfer the selected file name to the control display. Use the **Close** button to dismiss the display without selecting a file. When a file is selected in this manner, its directory then becomes the current search directory for the next file browse action.



Txfile #0 Transmit Load File

File: D:\Development\Executables\Server\Release\save\

D:\Development\Executables\Server\Release\save\

..
NexTxFile
save.txt
TxfileTest

Note: The file browser cannot navigate to a disk drive other than the drive where the server software is installed.

TxFile-5.2.3.2 File Types

Any type of data file may be used with varying degrees of success. Depending on the type of file, different modes of operation should be used. Currently this module only recognizes two basic file types.

Raw data:	Any file that contains data bytes only. It contains no additional bytes preceding or following the data bytes. An example of a raw data file is any log file created by the SIMSS Log Module without the log header option. Raw data files may be received from many other sources.
Log file:	Any file created by the SIMSS Log Module with the log header option enabled. The log header bytes contain data length and log time information.

A file may have fixed or variable length data.

Fixed data:	Every data entry in the file is the same size. When the Log Module creates fixed sized files, some entries may be truncated and others may be padded with zero bytes.
Variable data:	Data entries in the file may vary in size. This is usually the most flexible and efficient way to log data.

The following table summarizes read and transmission modes for the different file types and provides a map to applicable examples in the following sections.

File Type	Data Size	Read Modes	Transmit Modes	Examples
Raw data	Fixed	Offset	Manual mode	Example 1
Raw data	Fixed	Offset	Auto-Complete File	Example 2
Raw data	Fixed	Offset	Auto-Blocks	Example 3
Raw data	Varying	None apply	None apply	None apply, see note
Log file	Fixed or Varying	Log file	Manual mode	Example 1
Log file	Fixed or Varying	Log file	Auto-Complete File	Example 2
Log file	Fixed or Varying	Log file	Use Log file timing	Example 4

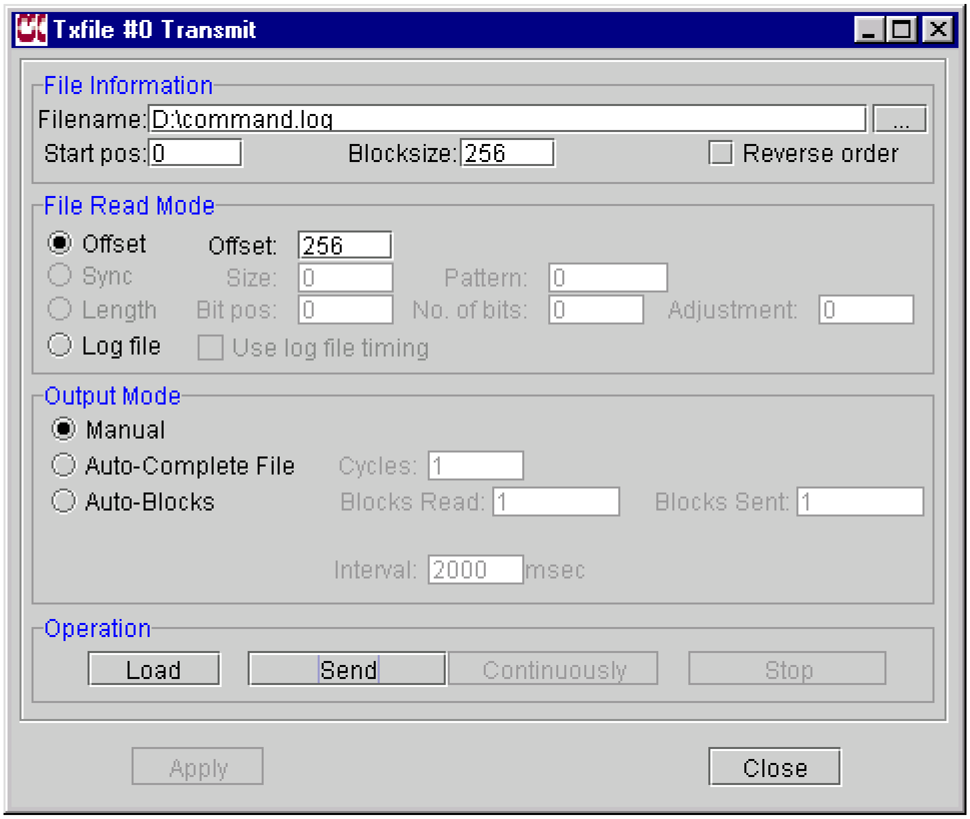
Note Since the Sync and Length read modes are not yet implemented, TxFile can not properly process raw files with varying size data. Every data byte can be read and transmitted using modes valid for fixed raw files, but the timing and grouping of the data may not be valid for the receiving system. These files should be converted to fixed data size files. When varying length data is recorded with the Log module for later TxFile playback, the log header option must be enabled.

The next five sections provide examples for using the different output modes. The steps given in each example are the minimum steps required. Review the parameter description table at the beginning of section 5.2.3 for more information.

TxFile-5.2.3.3 Example 1 Manual Mode Transmission

Step	Manual Mode Transmission Description
1	Enter the file name in the Filename field.
2	Select Manual Output Mode if it is not already on. If the Output Modes are not available for selection, click to select the Offset File Read Mode first.
3	For raw data files, click to set Offset File Read Mode and enter the fixed data size in bytes in the Blocksize and Offset fields. For log files, click to set the Log file option.
4	Click Apply button and note any error messages in the event message log. If necessary correct the file name and repeat this step until there are no errors.
5	Click Load button to read in the next entry of the file.
6	Click Send button to transmit the loaded entry one time.
	Repeat steps 5 and 6 as often as needed.

The following screen shows TxFile configured for manual output mode using a raw data file with entries of 256 bytes. Steps 1 through 4 (Apply) have been done. Steps 5 and 6 must be executed to transmit data.



Filename: D:\command.log
Start pos: 0 Blocksize: 256 □ Reverse order

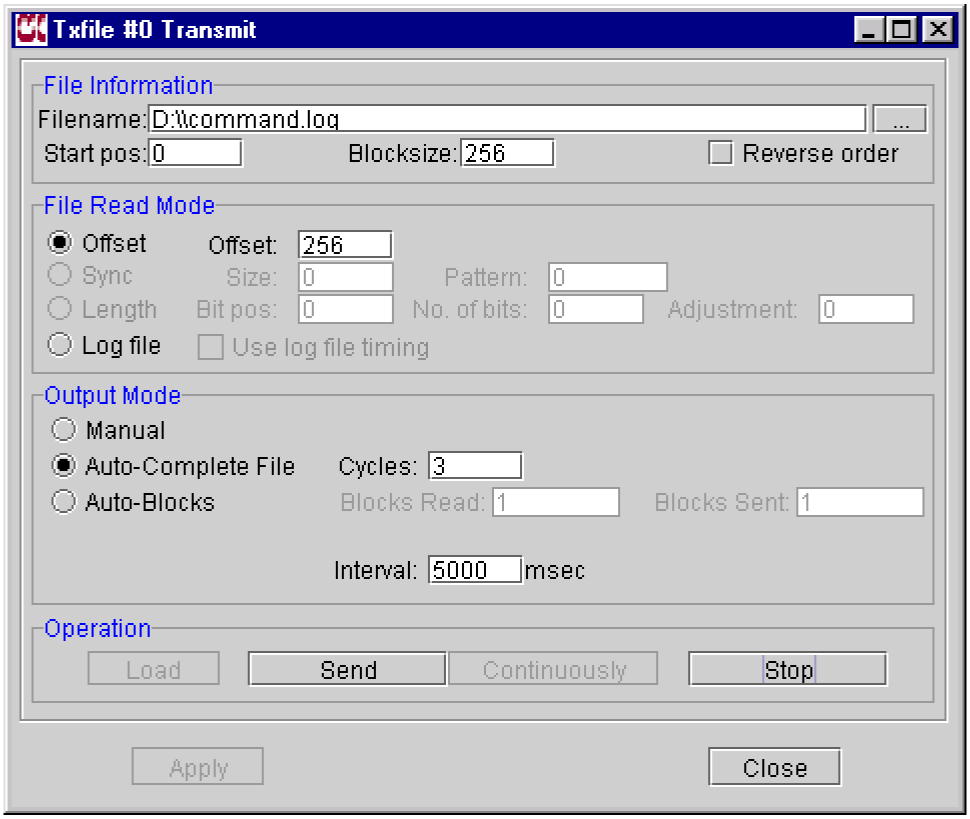
● Offset Offset: 256
○ Sync Size: 0 Pattern: 0
○ Length Bit pos: 0 No. of bits: 0 Adjustment: 0
○ Log file □ Use log file timing

● Manual
○ Auto-Complete File Cycles: 1
○ Auto-Blocks Blocks Read: 1 Blocks Sent: 1
Interval: 2000 msec

TxFile-5.2.3.4 Example 2 Auto-Complete File Transmission

Step	Auto-Complete File Transmission Description
1	Enter the file name in the Filename field.
2	Select Auto-Complete File in Output Mode region. If the Output Modes are not available for selection, click to select the Offset File Read Mode first.
3	For raw data files, select Offset File Read Mode and enter the fixed size in bytes in the Blocksize and Offset fields. For log files, click to set the Log file option.
4	Enter the desired timing between blocks in the Interval field. (1000 = 1 second)
5	For raw data files only, the Cycles field may be modified to specify the number of times to transmit the file. Set Cycles to zero for an indefinite number of times.
6	Click Apply button and note any error messages in the event message log. If necessary correct the file name and repeat this step until there are no errors.
7	Click Send button to start the automatic transmission. Transmission will stop when the file has been transmitted the specified number of cycles. Transmission does not automatically stop for raw files when the Offset field or the Cycles field is set to zero. Log files are only transmitted one time.
8	Click Stop button to stop the automatic transmission at any time.

The following screen shows TxFile configured for Auto-Complete File output mode using a raw data file with entries of 256 bytes. The entire file will be transmitted 3 times with 5 seconds between blocks. Steps 1 through 6 (Apply) have been done. A single click on the Send button will start the transmission.



Filename: D:\\command.log

Start pos: 0 Blocksize: 256 □ Reverse order

● Offset Offset: 256

○ Sync Size: 0 Pattern: 0

○ Length Bit pos: 0 No. of bits: 0 Adjustment: 0

○ Log file □ Use log file timing

○ Manual

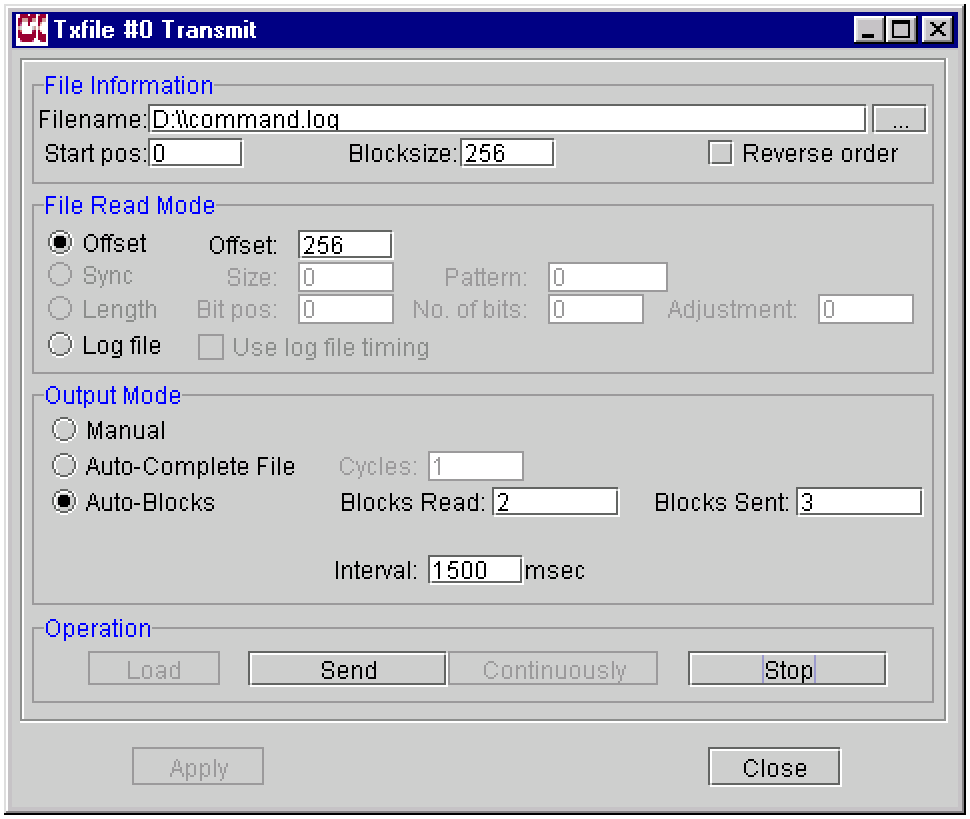
● Auto-Complete File Cycles: 3

○ Auto-Blocks Blocks Read: 1 Blocks Sent: 1

Interval: 5000 msec

TxFile-5.2.3.5 Example 3 Auto-Blocks Transmission (Raw Fixed Data Files Only)

Step	Auto-Blocks Transmission (Raw Fixed Data Files Only)
1	Enter the file name in the Filename field.
2	Select Auto-Blocks in the Output Mode region. If the Output Modes are not available for selection, click to select the Offset File Read Mode first.
3	Set Offset File Read Mode if not already set and enter the fixed data size in bytes in the Blocksize and Offset fields.
4	Enter the desired timing between blocks in the Interval field. (1000 = one second)
5	Enter the number of blocks to read in the Blocks Read field.
6	Enter the number of times to transmit each block in the Blocks Sent field. If this field is set to zero, the same block will be sent until the Continuously button is used to force loading of the next block.
7	Click Apply and note any error messages in the event message log. If necessary correct the file name and repeat this step until there are no errors.
8	Click Send to start the automatic transmission. If Blocks Read specifies a finite value, the transmission will stop after the specified blocks have been sent the specified number of times.
9	If the Blocks Read field was set to zero, click on Continuously button when desired to cause loading and sending of the next block.
10	Click Stop to stop the automatic transmission at any time.



Txfile #0 Transmit

File Information

Filename: D:\\command.log

Start pos: 0

Blocksize: 256

□ Reverse order

File Read Mode

● Offset Offset: 256

○ Sync Size: 0 Pattern: 0

○ Length Bit pos: 0 No. of bits: 0 Adjustment: 0

○ Log file □ Use log file timing

Output Mode

○ Manual

○ Auto-Complete File Cycles: 1

● Auto-Blocks Blocks Read: 2 Blocks Sent: 3

Interval: 1500 msec

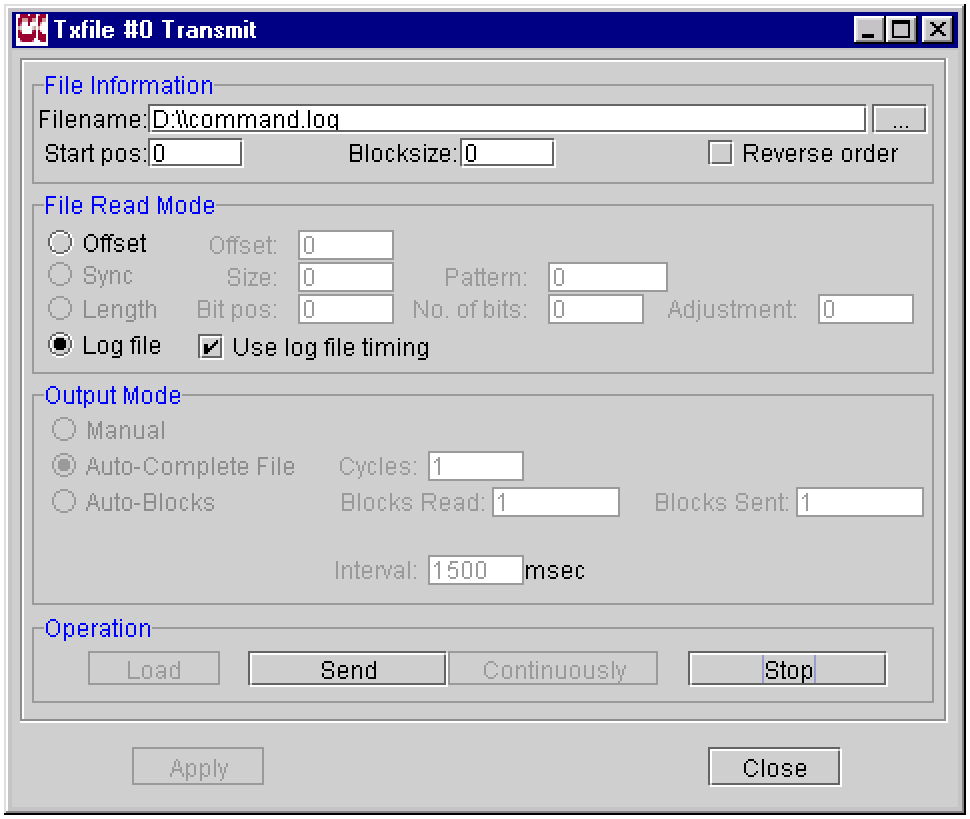
Operation

The above screen shows TxFile configured for Auto-Blocks output mode using a raw data file with entries of 256 bytes. The first 2 blocks of the file will be transmitted 3 times with 1500 milliseconds between blocks. Steps 1 through 7 (Apply) have been done. A single click on the Send button will start the transmission.

TxFile-5.2.3.6 Example 4 Use of Log File Timing (Log Files Only)

Step	Use of Log File Timing (Log Files Only)
1	Enter the log file name in the Filename field.
2	Select Auto-Complete File in Output Mode region. If the Output Modes are not available for selection, click to select the Offset File Read Mode first.
3	Click to select the Log file option.
4	Click to select the Use log file timing option. Relative timing between blocks will be calculated from the time tags in the log headers.
5	Click Apply and note any error messages in the event message log. If necessary correct the file name and repeat this step until there are no errors.
6	Click Send to start the automatic transmission of the file using log file timing. Transmission will stop when the file has been sent one time.
7	Click Stop to halt the automatic transmission before completion.

The following screen shows TxFile configured for Auto-Complete File mode using log file timing. Steps 1 through 5 (Apply) have been done. A single click on the Send button will start the transmission.



Filename: D:\\command.log

Start pos: 0

Blocksize: 0

□ Reverse order

○ Offset Offset: 0

○ Sync Size: 0 Pattern: 0

○ Length Bit pos: 0 No. of bits: 0 Adjustment: 0

● Log file ☑ Use log file timing

○ Manual

● Auto-Complete File Cycles: 1

○ Auto-Blocks Blocks Read: 1 Blocks Sent: 1

Interval: 1500 msec

TxFile-5.2.3.7 Example 5 Serial Transmission Mode

When TxFile is linked to a module requesting serial output, the transmission mode is automatically changed to serial transmission mode. When an **Automatic** output mode is selected, the module requesting serial data will control the timing of the transmissions. The operator may specify the file to read, read parameters and the number of times to transmit the file. Specification of log file timing or an interval will be ignored. Event messages will inform the operator when the mode is changed to serial, when transmission starts, when transmission stops, and each time the end-of-file is reached.

Note: Use of **Manual** mode with serial transmit mode is the same as with IP transmit mode. Refer to the previous Manual example for instructions.

Note: Should the operator want to override the serial timing and use the operator-specified interval or log file timing options, use the following directive to force the transmit mode to stay in IP mode. Then follow any of the other examples.

Set TxFileTransmitMode 2

Step	**Serial Transmission Mode**
1	Enter the file name in the Filename field.
2	Select Auto-Complete File in the Output Mode region. If the Output Modes are not available for selection, click to select the Offset File Read Mode first.
3	For raw data files, select Offset File Read Mode and enter the fixed size in bytes in the Blocksize and Offset fields. For log files, click to set the Log file option.

4	Set the Cycles field to the number of times to transmit the file. Set Cycles to zero to specify an indefinite number of times.
5	Click Apply and note any error messages in the event message log. If necessary correct the file name and repeat this step until there are no errors.
6	Click Send to start the serial transmission. Transmission will automatically stop when the file has been sent the specified number of times.
7	Click Stop to halt the serial transmission before completion.
8	Optionally click Send to resume or repeat the serial transmission. If the transmission was previously stopped, it will be resumed at the halted position in the file. If the end of transmission was reached, the original request will be repeated. If the Send button is not available, click on the Stop button first.

TxFile-5.3 About Display

Clicking the “About” option of the module pop-up menu requests a display that lists information about the module such as the numbers of input and output channels and whether the module accepts operator directives.

TxFile-6.0 Special Operating Instructions

- It is important to specify the appropriate read mode for the data file type (see TxFile-5.2.3.2 File Types). If a raw data file is read using the Log file mode, data bytes will be interpreted as log header fields and may cause the TxFile module to crash. If a log file is read using the Offset mode, the log header fields will be incorrectly transmitted as part of the data.

- When using the Auto-Complete File and Auto-Blocks Output modes, it is critical to set the Interval field properly. This field is in milliseconds. If a very short interval is used, an entire file may be transmitted multiple times in a brief period. In some configurations, the automatic transmission does not stop without operator intervention. This may overwhelm the resources of the sending system or that of the receiving system.

- There are several limitations when using Log files. When some of the following options are selected, they currently do not work and there are no error event messages.

  1) The Auto-Blocks output mode does not work with log files. (If Auto-Blocks is selected with use log file timing, TxFile processes the request as Auto-Complete File with log file timing.)

  2) When using the Auto-Complete File output mode with log files, the file is only transmitted once. It does not matter how the Cycles field is set.

  3) A starting offset (Start pos) cannot be used with log files. If specified, it will be ignored. The log file will be read from the beginning.

4) Reverse order should not be used with log files.

- The Auto-Blocks output mode does not work in Serial Transmit mode.

Generic VC Processor Module

VC Processor-1.0 Overview

The Generic CCSDS Virtual Channel (VC) Processor is used to monitor and verify virtual channels on a CCSDS data stream.

VC Processor-2.0 Inputs

The VC Processor module has a single input channel.

Channel	Description
1	Accepts external data stream of CCSDS data. Data must be synchronous and contain exactly one VC packet per data burst.

VC Processor-3.0 Outputs

The VC Processor module has three separate output channels. The output for each channel is user configurable (see below for configuration instructions).

Channel	Description
1, 2, 3	Sends user-defined channels to other modules.

VC Processor-4.0 Container Items

Container Name	Description
VCCRCCheck	Enable/disable CRC checking. (0 = disable, 1 = enable)
VCRSCheck	Enable/disable RS checking. (0 = disable, 1 = enable)
Interleave	The value for interleave. This is only used if RS checking is enabled.
Virtualfill	The value for virtual fill. This is only used if RS checking is enabled.
Displyarray	The status of each channel. This updates the colored boxes in the runtime display (0 = yellow, 1 = green, 2 = red).
Container Name for each VC. *XX* represents a decimal number (00 – 63)	**Description – each container item is specific to a single VC**
NumVCXX	The number of VCDU/Transfer Frames.
VCSeqErrsXX	The number of sequence errors.
VCCRCXX	The number of CRC errors.
VCRSErr	The number of RS errors.
VCUncorrRSErrXX	The number of uncorrectable RS errors. Note: this is not yet implemented.
VCvallOffset	The telemetry value starting at the offset byte.
OffsetLocation	The offset byte where telemetry extraction will begin.

VC Processor-5.0 Displays

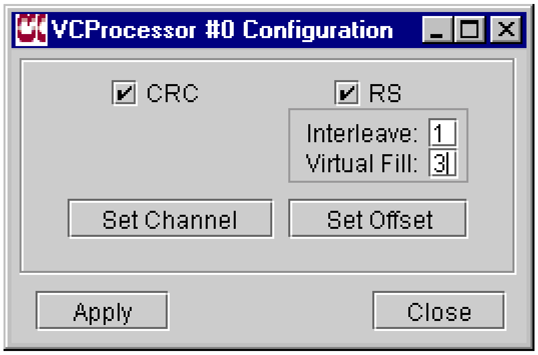
To access displays for a module, click in the center of the module in the project window. The following pop-up menu choices will appear. The “Remove” option can be used during project design to remove the module. The “Configure” option should be used prior to running the project and is unavailable at run-time. The Run-time option is available only when the project is running.

Module Pop-Up Menu Item	Description
Configure	Access the configuration display
Run-time	Access the Run-time menu for the module
Remove	Remove the module from the project
About	Display generic module information

VC Processor -5.1 Configure

There are three different configuration windows: Main Configuration, Set Channel and Set Offset. This module can also be re-configured during runtime.

VC Processor-5.1.1 Main Configuration Window



VCProcessor #0 Configuration

☑ CRC

☑ RS

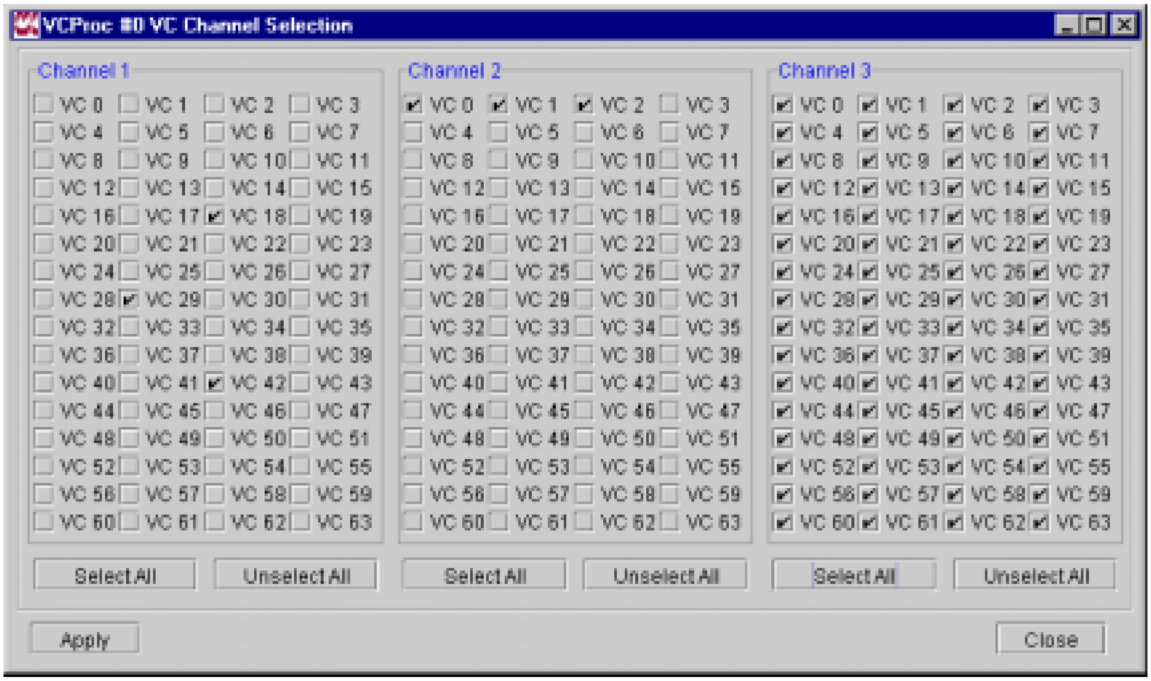
Interleave: 1

Virtual Fill: 3

Field Name	Description
CRC	Enable/disable CRC check.
RS	Enable/disable RS encoding check.
Interleave	Only accessible when doing RS check. Input is a decimal integer.
Virtual Fill	Only accessible when doing RS check. Input is a decimal integer.
Set Channel	Set virtual channels to transmit on each of the three output channels.
Set Offset	Set the start byte location for display value (one per virtual channel).

This is the main configuration window. CRC and RS checking is enabled/disabled from this window. The two other configuration windows are accessible from this window.

VC Processor-5.1.2 Set Channel



VCProc #0 VC Channel Selection

Channel 1

☐ VC 0 ☐ VC 1 ☐ VC 2 ☐ VC 3
☐ VC 4 ☐ VC 5 ☐ VC 6 ☐ VC 7
☐ VC 8 ☐ VC 9 ☐ VC 10 ☐ VC 11
☐ VC 12 ☐ VC 13 ☐ VC 14 ☐ VC 15
☐ VC 16 ☐ VC 17 ☑ VC 18 ☐ VC 19
☐ VC 20 ☐ VC 21 ☐ VC 22 ☐ VC 23
☐ VC 24 ☐ VC 25 ☐ VC 26 ☐ VC 27
☐ VC 28 ☑ VC 29 ☐ VC 30 ☐ VC 31
☐ VC 32 ☐ VC 33 ☐ VC 34 ☐ VC 35
☐ VC 36 ☐ VC 37 ☐ VC 38 ☐ VC 39
☐ VC 40 ☐ VC 41 ☑ VC 42 ☐ VC 43
☐ VC 44 ☐ VC 45 ☐ VC 46 ☐ VC 47
☐ VC 48 ☐ VC 49 ☐ VC 50 ☐ VC 51
☐ VC 52 ☐ VC 53 ☐ VC 54 ☐ VC 55
☐ VC 56 ☐ VC 57 ☐ VC 58 ☐ VC 59
☐ VC 60 ☐ VC 61 ☐ VC 62 ☐ VC 63

Channel 2

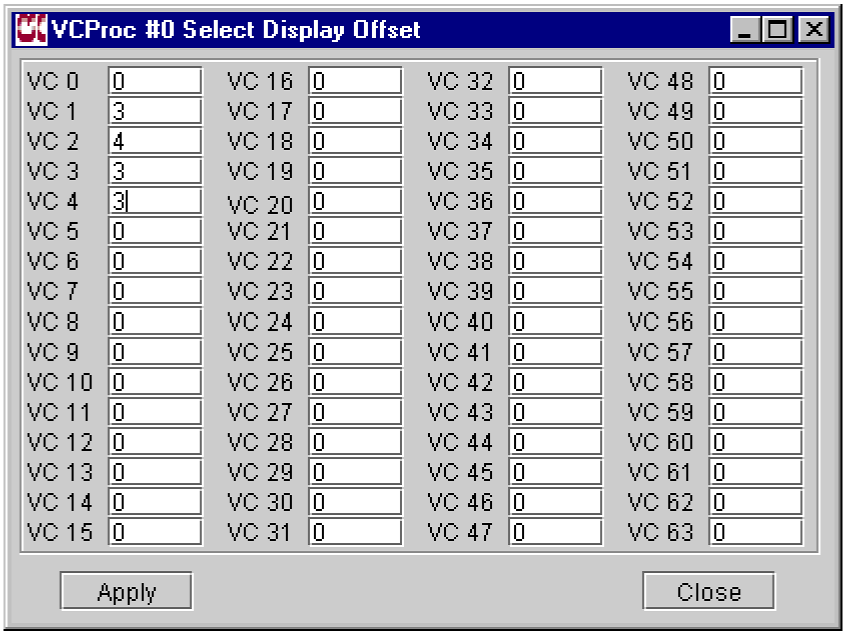
☑ VC 0 ☑ VC 1 ☑ VC 2 ☐ VC 3
☐ VC 4 ☐ VC 5 ☐ VC 6 ☐ VC 7
☐ VC 8 ☐ VC 9 ☐ VC 10 ☐ VC 11
☐ VC 12 ☐ VC 13 ☐ VC 14 ☐ VC 15
☐ VC 16 ☐ VC 17 ☐ VC 18 ☐ VC 19
☐ VC 20 ☐ VC 21 ☐ VC 22 ☐ VC 23
☐ VC 24 ☐ VC 25 ☐ VC 26 ☐ VC 27
☐ VC 28 ☐ VC 29 ☐ VC 30 ☐ VC 31
☐ VC 32 ☐ VC 33 ☐ VC 34 ☐ VC 35
☐ VC 36 ☐ VC 37 ☐ VC 38 ☐ VC 39
☐ VC 40 ☐ VC 41 ☐ VC 42 ☐ VC 43
☐ VC 44 ☐ VC 45 ☐ VC 46 ☐ VC 47
☐ VC 48 ☐ VC 49 ☐ VC 50 ☐ VC 51
☐ VC 52 ☐ VC 53 ☐ VC 54 ☐ VC 55
☐ VC 56 ☐ VC 57 ☐ VC 58 ☐ VC 59
☐ VC 60 ☐ VC 61 ☐ VC 62 ☐ VC 63

Channel 3

☑ VC 0 ☑ VC 1 ☑ VC 2 ☑ VC 3
☑ VC 4 ☑ VC 5 ☑ VC 6 ☑ VC 7
☑ VC 8 ☑ VC 9 ☑ VC 10 ☑ VC 11
☑ VC 12 ☑ VC 13 ☑ VC 14 ☑ VC 15
☑ VC 16 ☑ VC 17 ☑ VC 18 ☑ VC 19
☑ VC 20 ☑ VC 21 ☑ VC 22 ☑ VC 23
☑ VC 24 ☑ VC 25 ☑ VC 26 ☑ VC 27
☑ VC 28 ☑ VC 29 ☑ VC 30 ☑ VC 31
☑ VC 32 ☑ VC 33 ☑ VC 34 ☑ VC 35
☑ VC 36 ☑ VC 37 ☑ VC 38 ☑ VC 39
☑ VC 40 ☑ VC 41 ☑ VC 42 ☑ VC 43
☑ VC 44 ☑ VC 45 ☑ VC 46 ☑ VC 47
☑ VC 48 ☑ VC 49 ☑ VC 50 ☑ VC 51
☑ VC 52 ☑ VC 53 ☑ VC 54 ☑ VC 55
☑ VC 56 ☑ VC 57 ☑ VC 58 ☑ VC 59
☑ VC 60 ☑ VC 61 ☑ VC 62 ☑ VC 63

This window is the output channel configuration window. Any number of virtual channels can be transmitted over any of the three module output channels. The checked VCs will be transmitted on the specified channel. In the above picture, module output channel 1 will transmit VCs 18, 29, and 42.

VC Processor-5.1.3 Display Offset



VCProc #0 Select Display Offset

VC 0	0	VC 16	0	VC 32	0	VC 48	0
VC 1	3	VC 17	0	VC 33	0	VC 49	0
VC 2	4	VC 18	0	VC 34	0	VC 50	0
VC 3	3	VC 19	0	VC 35	0	VC 51	0
VC 4	3	VC 20	0	VC 36	0	VC 52	0
VC 5	0	VC 21	0	VC 37	0	VC 53	0
VC 6	0	VC 22	0	VC 38	0	VC 54	0
VC 7	0	VC 23	0	VC 39	0	VC 55	0
VC 8	0	VC 24	0	VC 40	0	VC 56	0
VC 9	0	VC 25	0	VC 41	0	VC 57	0
VC 10	0	VC 26	0	VC 42	0	VC 58	0
VC 11	0	VC 27	0	VC 43	0	VC 59	0
VC 12	0	VC 28	0	VC 44	0	VC 60	0
VC 13	0	VC 29	0	VC 45	0	VC 61	0
VC 14	0	VC 30	0	VC 46	0	VC 62	0
VC 15	0	VC 31	0	VC 47	0	VC 63	0

Each VC will display a 32-bit telemetry value in the runtime Status window (5.2.1). The byte offset (first byte to display) is configured in this window. The first byte in a VC is byte zero. For example, an offset value of 3 will display bytes 3, 4, 5, and 6 in the Status window when the module is running.

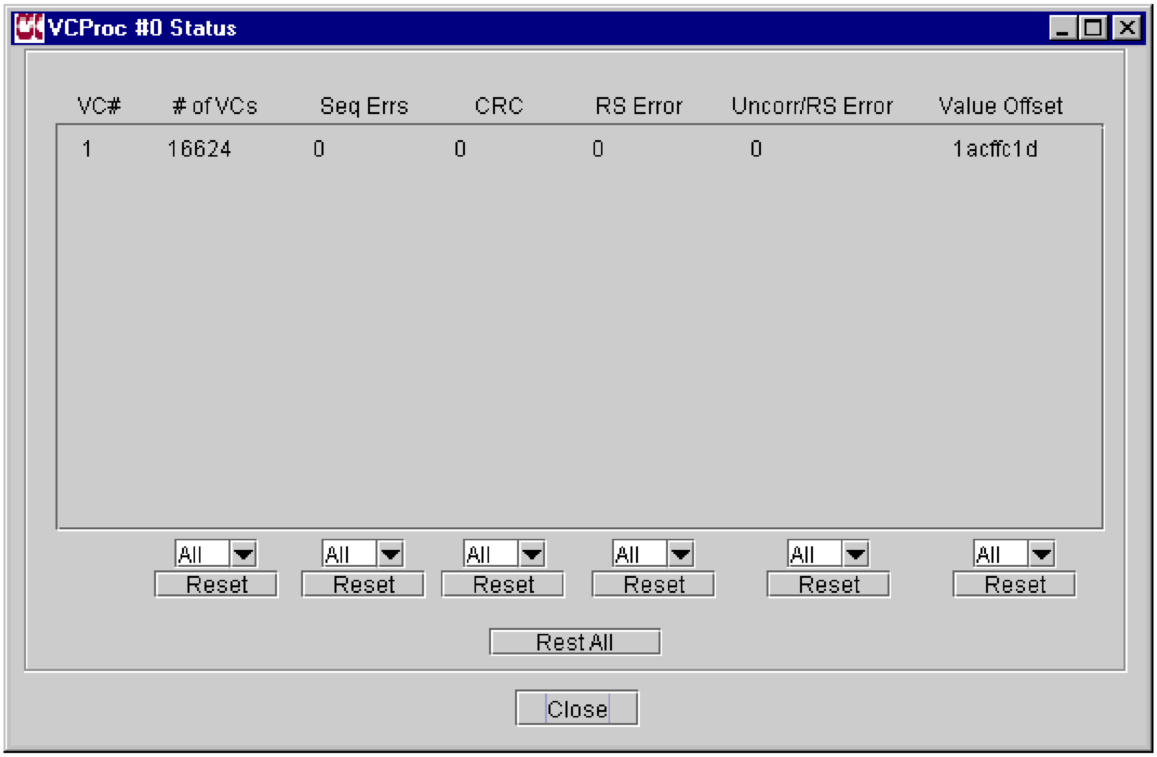
VC Processor-5.2 Run

During runtime, the VC Processor module will perform an analysis of the CCSDS data stream going through it. When the module encounters a new VC in the stream, it will inform the user through status windows and report on the status of each active VC.

The Run-time menu for the VC Processor contains the following five items.

Run-time Menu Item	**Description**
Show Status	Display the status of each active VC.
Display VC	A “quick” display of all VCs.
ReConfigure	Allows user to reset configuration variables.
Resume	Resumes module.
Pause	Pauses module.

VC Processor-5.2.1 Show Status



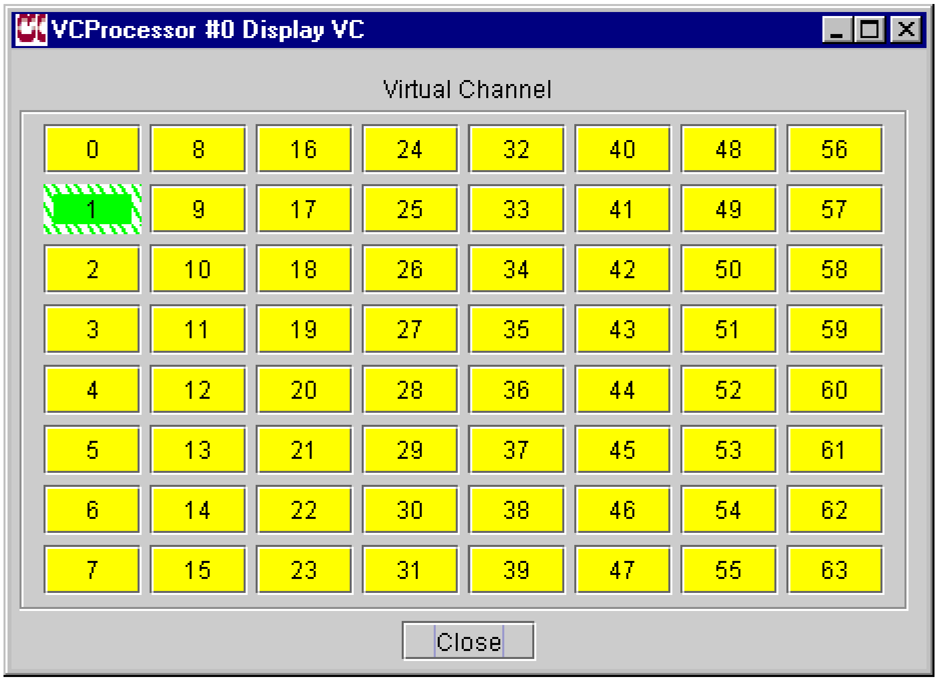
VC#	# of VCs	Seq Errs	CRC	RS Error	Uncorr/RS Error	Value Offset
1	16624	0	0	0	0	1acffc1d

This window shows the status of each active VC.

Status Column	**Description**
VC#	The VC ID (0 – 63).
# of VCs	The number of VCDU/Transfer Frames received for this virtual channel.
SeqErrs	The number of sequence errors encountered for this virtual channel.

CRC	The number of CRC errors encountered for this virtual channel. This will be zero if CRC checking is not enabled.
RS Error	The number of RS errors encountered for this virtual channel. This will be zero if RS checking is not enabled
Uncorr/RS Error	This functionality is not implemented yet.
Value Offset	The 32 bit value taken from telemetry. The location to start extracting data from is found in the configuration window

VC Processor-5.2.2 Display VC



VCProcessor #0 Display VC

Virtual Channel

0	8	16	24	32	40	48	56
1	9	17	25	33	41	49	57
2	10	18	26	34	42	50	58
3	11	19	27	35	43	51	59
4	12	20	28	36	44	52	60
5	13	21	29	37	45	53	61
6	14	22	30	38	46	54	62
7	15	23	31	39	47	55	63

This is a “quick” display of the status of all VCs. The following color code applies:

Yellow	Inactive VC
Green	Active VC with no errors
Red	Active VC with errors

All VCs start out as inactive. When a VC is encountered in the data stream, the corresponding box turns green or red, depending on quality of data. If a box turns red, the user can click on it, turning it yellow (inactive). A VC box will not change from red to any other color unless the user clicks on it.

VC Processor-5.2.3 ReConfigure

This option allows the user to re-configure the VC Processor module during run-time. Neither the module nor the project needs to be stopped in order to reconfigure. Refer to the configuration section (5.1) for details.

ACRONYMS

APID	Application Identifier
BCH	Bose-Chaudhuri-Hocquenghem
CADU	Channel Access Data Unit
CCSDS	Consultative Committee on Space Data Systems
CLCW	Command Link Control Word
CLTU	Command Link Transmission Unit
COP	Command Operations Procedure
COTS	Commercial, off-the-shelf
CSC	Computer Sciences Corporation
CUC	CCSDS Unsegmented Time Code
EDOS	EOS Data and Operations System
EDU	EDOS Data Unit
EGS	EOS Ground System
EMOS	EOS Mission Operations System
EOSGS	SIMSS module for the EOS Ground Station
EOS	Earth Observing System
EOSDIS	Earth Observing System Data and Information Systems
EPGS	EOS Polar Ground Stations
ESH	EDOS Service Header
ETS	EOSDIS Test System
FARM	Frame Acceptance and Reporting Mechanism
FTP	File Transfer Protocol
GMH	Ground Message Header
GMT	Greenwich Mean Time
GUI	Graphical User Interface
GS	SIMSS Ground Station Module
ICD	Interface Control Document
IP	Internet Protocol
MPS	Multimode Portable Simulator (legacy simulator for AM-1)
MPS/Aura	Common name for the SIMSS/Aura Simulator
NT	New Technology
PC	Personal Computer
PDB	Project Database
SC	SIMSS Spacecraft Module
SCAura	SIMSS Spacecraft Module with Aura extensions
SIMSS	Scalable Integrated Multimission Simulation Suite
SIMSS/Aura	SIMSS-based Aura Simulator (also known as MPS/Aura)
TCP	Transmission Control Protocol
TIE	Transponder Interface Electronics
UDP	User Datagram Protocol
VCDU	Virtual Channel Data Unit
VCID	Virtual Channel Identification